UNIVERSITY OF HERTFORDSHIRE

Faculty of Engineering & information science

MSc. in ELECTRONICS

(DIGITAL SYSTEMS**)**

Project Report

A FLEXIBLE DEVELOPMENT PLATFORM FOR INTERACTIVE ROBOTS

Gulwinder Singh

April 2003

University of Hertfordshire

"I certify that the work submitted is my own and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged(ref. UPR 17.1,section on cheating and plagiarism)

Signed: ……"

University of Hertfordshire

ABSTRACT

This report is a detailed description of the work carried out in order to design a Motorola 68000 microprocessor based flexible development platform for interactive robots in term of its applications and usability. Due to tremendous development in sensor technology the robots become more flexible and versatile by applications of their sensors. As Lego Mindstorms allows to build a robot that works in a specific environment but the main objective of this project is to develop an application that allows interaction with a robot from the outside world where motors are controlled according to the physical parameters of the different sensors easily.

The aim of this report is to demonstrate the main function and application of the interactive robot by using embedded chips and c program. The reasons for choosing a MC68000 application board and 'C' language to develop this system are included in this report. Various work stages and development of the project is discussed. Also included is a methodical description of the construction and testing of the sensors involved in the project and finally a conclusion, which discusses all the important aspects of this successfully completed project. Any subsequent reader should be able to gain valuable knowledge from this report.

University of Hertfordshire

Acknowledgements

During this project, there have been many people that have given me their valuable time and support, so I would like to give them special thanks. The help of all such people is gratefully acknowledged.
In particular, the following people at the University of Hertfordshire deserve my gratitude.

First of all, I would like to express my gratitude to **Dr. David Lee (Project Supervisor)** for his invaluable devotion and supervision throughout the period of the project. His guidance and considerable experience are greatly appreciated .His advise and continuous support had certainly helped me, particularly during the difficulties are encountered.

Mr Tony Crook (Technical Manager) for assisting with many practical decisions and providing all the components which I required for my project without any delay .

Mr. Ian Munro, researcher at the University of Hertfordshire, for advice and assistance on the project.

Mr Stephen Passmore (Technician) for providing assistance with access to the microprocessor laboratory and making available the necessary equipment when needed.

Finally, this project is dedicated to my family who provides me a opportunity to study my Master's degree in U.K and supports me all the moment . This project and report would not have been completed without the encouragement and love which I received from my family.

University of Hertfordshire

CONTENTS

I. Certificate

II. Abstract

III. Acknowledgements

IV. Contents

V. List of Figures and Tables

CHAPTER 1: INTRODUCTION

1.1 Preview ---------- 1

1.2Project Background.----------1

1.3 Project Aims And Objectives.----------3

1.4Project Overview----------4

1.5 Chapter Organisation. ----------6

1.5 Summary.----------8

CHAPTER 2: SENSORS AND MOTORS

2.1 Introduction.----------9

2.1.1 Sensor Operation Principle----------10

2.1.2 Types Of Sensors----------11

2.1.3 Sensors Output Values----------12

2.1.4 Evaluating a Sensor----------12

2.1.5 Advantages and Disadvantages of the Sensors----------16

2.2 Lego mindstorm and DCP Sensors.----------17

2.3 Choice of the Sensors----------19.

2.4 Choice of the Motors.----------20

University of Hertfordshire

2.4.1 Control of Motor----------21
2.5 Summary.----------22

CHAPTER 3 : LITERATURE RESEARCH

3.1 Literature Research.----------23
3.2Hardware Research .----------23

3.3 Software Research.----------24
3.4 Summary----------24

CHAPTER 4 : C AND 68000

4.1 Motorola 68000.----------25
4.2 FLIGHT.-68K Board----------28
4.3. Choosing 'C' language for programming----------30
4.4 Summary----------31.

CHAPTER 5 : HARDWARE DESIGN

5.1 Designing and Testing of Sensors----------32
5.1.1 Voltage Sensor----------
33.
5.1.2 Switch Sensor----------
35.
5.1.3 Light Sensor.----------36
5.1.4 Sound Sensor----------37
5.2 Analogue to Digital Converter----------38
5.3 Bus Expander----------41
5.4 I/O Buffer Board ----------42
5.5 Stepper Motor Controller Board ----------43
5.6 Address Decoding ----------45

University of Hertfordshire

5.7Experimental Setup of the Flexible Development Platform----------47
5.8 Summary.----------49

CHAPTER 6 : SOFTWARE DESIGN

6.1 Introduction----------50
6.2 Software Program----------51
6.3 Summary----------55

CHAPTER 7 : FUNCTIONAL TEST RESULT

7.1 I/O Buffered Test. ----------56
7.2 Single Input ADC Test----------57.
7.3 Multiple Input ADC Test ----------57..
7.4 Summary.----------59

CHAPTER 8 : PROJECT MANAGEMENT

8.1 Overview.----------60
8.2 Time Management ----------60
8.3 Summary.----------64

CHAPTER 9: CONCLUSION AND FUTHER DEVELOPMENT

9.1 Introduction.----------65
9.2 Conclusion. ----------65
9.3 Further Work And Development ----------66.

***REFERENCE** ----------68-

***BIBLOGRAPHY**----------69

***APPENDIX**

APPENDIX A.1: BlockDiagram of the FLIGHT -68K Board----------70
APPENDIX A.2: specifications of the FLIGHT -68K BOARD ----------71
APPENDIX B.1: Program Code for Whole Flexible Development Platform ----73
APPENDIX B.2 . Program code for testing I/O buffered board.----------76

University of Hertfordshire

APPENDIX B.3 Program Code for Testing Single Input of the ADC Board------78

APPENDIX C : Registers of PI/T 680230----------80
APPENDIX D : Stepper Motor Board Circuit Diagram----------81
APPENDIX E : I/O Buffered Board Internal Circuit Diagram----------82
APPENDIX F : Bus Expander Board Internal Circuit Diagram----------83

APPENDIX G.1 : ADC Board Internal Circuit Diagram----------84

APPENDIX G.2 : ADC Board Internal Circuit Diagram----------85

APPENDIX H.1: Connection Diagram Of Op-Amp 741----------86

APPENDIX H.2: Data Sheet Of Op-Amp 741----------87

APPENDIX I: Data Sheet Of LDR Used in Light Sensor----------88

APPENDIX J: Data Sheet Of Microphone Used in Sound Sensor----------89

APPENDIX K Practical laboratory set up of the project----------90

University of Hertfordshire

LIST OF FIGURES AND TABLES

Figure 1.1: - RCX LEGO Brick……3
Figure1.2: - Specified diagram of all the blocks of the project system……6
Figure 2.1:- Some types of Energy forms converted by the Sensors…….. 1
Figure 2.2- Principle of the Sensor……..10
Figure 2.3:- Different TypesSignals……10
Figure 2.4 General purpose analogue interface circuit……19
Figure 2.5 Function of pins of DCP sensors……21
Figure 2.5 – Square Waveform……23
Figure 4.1 68000 function codes……...29
Figure 4.2 Logical arrangements of the 68000 pins……29
Figure 4.3: Connection of PC and FLIGHT……30
Figure 5.1 Voltage Amplifier Using Op-Amp 741…… 34
Figure 5.2 – Simple Voltage Sensor Circuit……35
Figure 5.3: Graphical representation of output of Voltage sensor……..36
Figure 5.4 – Switch Sensor……..37
Figure 5.5: LDR Resistance Vs Output Voltage Of light Sensor……38
Figure 5.6: Graphical representation of the light sensor output……..39
Fig ure5.7 Sound Sensor Circuit Diagram ……40
Figure 5.8: ADC Board……...42
Figure5.9: Bus Expander Board……....43
Figure 5.10:- I/O Buffered Board …..……45
Figure: 5.11 Stepper Motor Controller Board…….....46
Figure 5.12: Timing Diagram of Different Modes of Stepper Motor..…….....47
Figure 5.13 .Timing Diagram of Segment and Index pulse[18]……47
Figure 5.14 Decoder Used for Address decoding……49
Figure 5.15 : Experimental setup of the whole system ……..50
Figure 6.1 : Working of the software program……57
Figure 7.1 Output waveform at Port B by Voltage sensor…….....61
Figure 8.1 Initial and Final time plan Gantt chart……..66

University of Hertfordshire

Table 2.1 Summary Input Table……20
Table 4.1 MC68230 PI/T addresses of F68k……..31
Table 5.1 Output of the Voltage Sensor……36
Table 5.1 Analogue Signal Converted into Discrete Values……41
Table 5.2: Data for selecting ADC Input Pin……42
Table 5.3 Address Decoding Table……48
Table 8.1 : Time table of the important steps of the project……63

University of Hertfordshire

CHAPTER 1

INTRODUCTION

1.1 Preview

Today most robots designed to be a helping hand. They help people with tasks that would be difficult, unsafe, or boring for a real person to do alone. At its simplest, a robot is machine that can be programmed to perform a variety of jobs, which usually involve moving or handling objects. Robots can range from simple machines to highly complex, computer-controlled devices. The robots of the movies, such as C-3PO and the Terminator are portrayed as fantastic, intelligent, even dangerous forms of artificial life. [9].

However, robots of today are not exactly the walking, talking intelligent machines of of movies, stories and our dreams. Today, we find most robots working for people in factories, warehouses, and laboratories. In the future, robots may show up in other places: our schools, our homes, even our bodies. Robots have the potential to change our economy, our health, our standard of living, our knowledge and the world in which we live. As the technology progresses, we are finding new ways to use robots. Each new use brings new hope and possibilities, but also potential dangers and risks.

The Robot in this project means any kind of control machine, for example: transport machine for product line, robot arm in car doing painting, welding, loading and unloading, inspection and assembly.

1.2 Project Background

The objective of this project is to design a controller to control the sensors and actuators(motors) on a robot by interfacing with MC68000 which is connected with PC via RS232. This means any other computer with an RS232 interface will be able to control a robot using MC68000.

University of Hertfordshire

There is a similar project called "*The Lego Mindstorms Robots* "from which I take a basic idea for my project. This system has limited inputs and also used expensive components like sensors, connectors and motors .So many students were working on this Lego Mindstorms robot but nobody tried to make their own flexible system especially using MC6800 therefore this design is tried to achieve in this project. The brief description of this project is necessary before starting this project.

Actually Mindstorms is the name of a product line sold by the Lego group that is suggested for children above 11 years old.

Figure 1.1: - RCX LEGO Brick

The LEGO® MINDSTORMS™ **Robotics Invention Systems™ (RIS)** is a powerful new technology. It includes a programmable microcomputer called **Robotics Command System (RCX)**, which is built into a LEGO brick and able to execute the programs downloaded from a

University of Hertfordshire

computer. The **RCX (Robotics Command System)** is a programmable LEGO brick. It has three sensor (input) ports, three output ports, and four control buttons, an LCD display, and an infrared transmitter as shown in figure 1.1.

It also has a microprocessor to process programs, internal memory to store firmware and programs, and a built-in speaker to produce beeps and tones. The light sensor, temperature sensor and touch sensors provide inputs to the RCX from the environment. The RCX can then process the data to control the motors (turn on/off or forward/reverse). The sensors, motors and other components included, in combination with the LEGO bricks, allow users to build a blinding array of clever devices. User can construct and program robots in such a way that these robots can be trained to move, act, and think like their own. [1]

This "smart brick" can process over 1000 commands per second. The RCX uses an 8-bit Hitachi H8/3297 microcontroller running at 16MHz. It has 16K of ROM, 512 bytes of SRAM (for firmware) and 32K of external SRAM for the programs [2]

1.3 Projects Aims and Objectives:

The aims and objectives of this project have been proposed in the early stage of the project work with the elaboration of the feasibility study report

The key objectives that need to be considered and accomplished in order to achieve the aim of designing a working platform which satisfies the design brief are listed below:

- To learn and understand the function of the Lego Mindstorms robots to develop flexible development platform for interactive robots.
- To study and research on different type of sensors used in Robots.
- To interface the sensors with the 68000 microprocessor as input with motor, which is connected at output.
- To learn hardware troubleshooting learn self-discipline and time management.
- Learn and research about the MC68000 microprocessor and the Flight 68000 training systems available within the university, in particular how to use and interface to them.
- Design and build circuitry to interface to the MC68000 training system and software program suitable for operating the completed hardware
- To learn how to communicate and discuss with supervisor and people relevant about the project.
- To gain valuable knowledge and Experience in the Microprocessor System Design, Electronic Sensors, Robotics ,C programming and Project Management areas.

University of Hertfordshire

The main objective of the project is to obtain the best suitable sensors in term of cost and accuracy to design this flexible platform. Some experience and new knowledge have been gained during the project research, assembling and interfacing the sensor with RCX. Besides that, it helps to improve technical skills and to solve problems more efficiently. Most important of all is the technical apply to the project while assembling the design.

The following **sub-objective** were achieved

- To build, design and create a new hardware by which any sensor (input device)interface with 68000 to control the motor(or any equipment according to requirement) which is connected at output.
- To Implement the program to simulate the sensors (input devices) and output devices with FLIGHT-68K using C programming language.
- To make the system as much as flexible so that any number of inputs and outputs are connected according to the application.
- To write control program using different type of sensors in C Language.
- To test the accuracy of the sensors when interfaced with Motorola 68000 via 68230 Parallel Interface/Timer on FLIGHT-68K board so that it performs more like a robot.

1.4 Project overview

To describe the project overview, the general block diagram overall system is shown in the figure 1.2 which consist of :

- A platform of different sensors at the input.
- A microprocessor FLIGHT-68K training board connected with PC via RS232 cable.
- Stepper motors which are controlled according to the physical parameters of the sensors.
- Designed circuitry to interface all these sensors and actuators with MC68000.

University of Hertfordshire

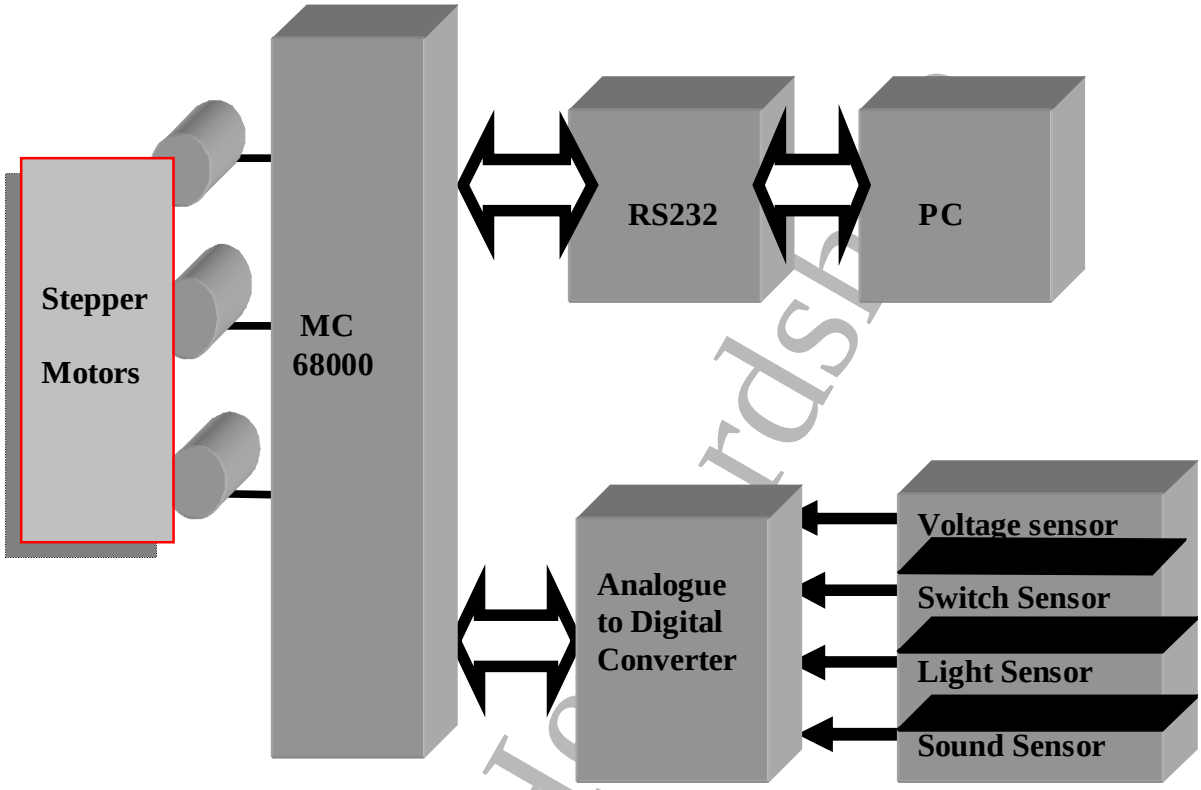


Figure 1.2: - Specified diagram of all the blocks of the project system

The project will be based on the 68000 development board and C program.
Explanation of all the parts of the system is given in the following chapters. It is true to say that this system is only as good as its sensors. Sensors estimates the environmental changes in term of analogue output, depend on its application. This output data converted into suitable real value and stored in memory by interfacing with microprocessor. Analysis of this stored data can be done to control the real world applications like speed of the stepper motor is controlled in term of the temperature of water by using temperature sensor here.

Some possible current application of this flexible system includes:

Gas Industry-	Gas Flow
	Pressure and Telemetry outstations
	Temperature control
Water industry -	River, Reservoir and Borehole Water Level
	Water Quality Parameters
	Rain Gauging
	Pump Duty Monitoring

University of Hertfordshire

Low Power Telemetry

Other Applications- Intelligent Children Toys
Energy Management
Traffic Counting
Industrial plant Monitoring
Pollution and Environmental Studies
Offshore Marine Systems
Air Quality Monitoring

1.5 Chapter Organisation

The report is targeted both at the experienced electronic engineer who is and is not a specialist in the field of sensors and developing flexible platform for different applications including MC68000 interfacing. The former can start from Chapter 5 directly to understand all the experimental results and electronic parts. The latter may study the whole report to learn the theory parts as well electronics parts. The following section contains a brief summary of all chapters including in this report.

Chapter 1 - Introduction

This chapter describes the project overview, its background theory and the report to the reader. It contains brief overview, aims and objectives of the project with its some applications and how the report is organized.

Chapter 2 - Sensors and Motors

This chapter covers the information about the basic theory behind the sensors, with their advantages and disadvantages. It describes the technologies (sensors and motors) available to do the project. Here are discusses several possible implementation options. It also include is a discussion about the advantages and disadvantages of the technology available and why it was chosen a certain one to develop the current application.

Chapter 3- Literature research

Summary of the research carried out for the project in its hardware and software field, and conclusion drawn from this, culminating in the project design specification.

University of Hertfordshire

Chapter 4- C and Motorola 68000

It contains the brief summary of the MC68000 and FLIGHT 68000 Microprocessor training system board with the reasons for choosing 'C' language for programming.

Chapter 5- Hardware design

A detailed description of the design of the different types of the sensors is given in this chapter .It also includes the various circuits using for interfacing of flexible system with MC68000 and details of memory mapping.

Chapter 6- Software design

This chapter presents the software details used in this project.

Chapter 7- Functional test results

This chapter includes the testing and troubleshooting of various circuits used for interfacing with MC68000 and the experimental setup of the whole system.

Chapter 8- Project management

It consists the time management for successful completion of the project with the omparison of the initial and final time plan Gantt chart.

Chapter 9- Conclusion and Further development

It contains all vital observations and conclusions of the project and some recommendations are also suggested for future development of this project.

1.6 Summary

This chapter includes the project background and the main aims and objectives of this project. An overview of what a project system consists of follows with some typical application and finally the summaries of chapter organisation of this report are stated.

University of Hertfordshire

University of Hertfordshire

CHAPTER 2

SENSORS AND MOTORS

2.1 Introduction

This chapter covers an introduction to basic of the sensors and motors and is not required reading for specialists in the theory of sensors fundamentals.

"A transducer is a device that converts input energy into output energy, the latter usually differing in kind but bearing a known relationship to input". Sensors form a small subset of the group of transducers and are defined as follows:

A sensor is a transducer which is capable of converting a physical non-electrical input quantity into an electrical output quantity and of processing it, in accordance with a given algorithm, to provide an output suitable for interfacing to a process control system such as computer. Many measuring and sensing devices, as well as loudspeakers thermocouples, microphones, and phonograph pickups, may be termed transducers.[14]

Any data a computer gets that has not been entered by a human user or already been put into digital form in another computer enters the computer from a sensor or transducer. The function of these sensors is to convert some measured physical quantity (pressure, temperature, fuel tank level, flow rate, position, velocity, acceleration, chemical concentration, etc.) into a voltage signal that can then be digitized with the usual analogue-to-digital (A/D) techniques. The technologies of sensors and signal conditioning (processing the sensor output to make it compatible with the computer) are two of the key drivers of progress in computer-based instrumentation, control and automation sensors convert different type of energy into the electrical energy as shown in figure 2.1.

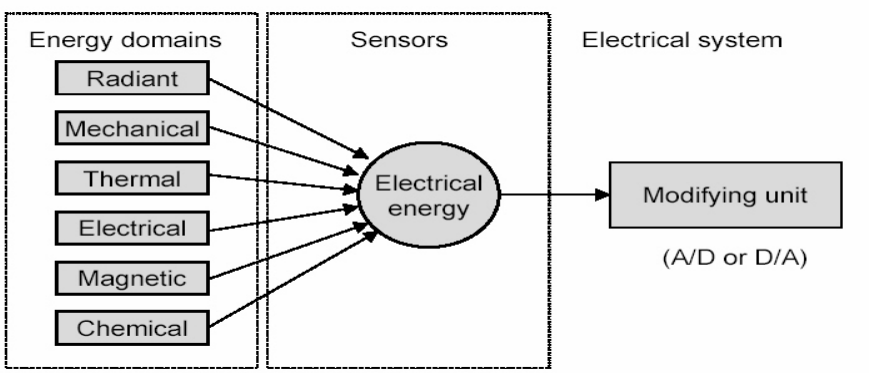


Figure 2.1:- Some types of Energy forms converted by Sensors.

2.1.1 Sensor Operational Principle

Most of the sensors operational principles is the same as the electromagnetic (radar, laser etc.) and active acoustic sensing

Univ f Hertfordshire

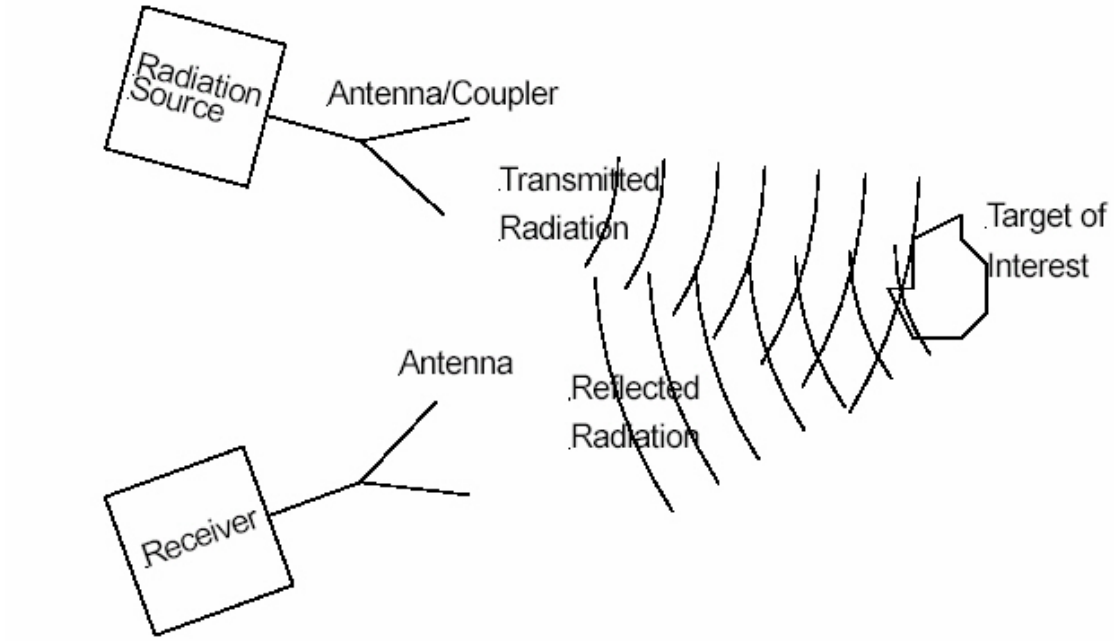


Figure 2.3 Principle of the sensor

The source of radiation is fed to a transmit antenna, tuned to the characteristics of the target (sometimes) , Matched to the impedance of the medium to maximise coupling and efficiency and radiated by a directional antenna to increase the energy on the target. These radiations strike at the target of interest and emit back from which a small percentage of power enters the receiver antenna and converted into the electrical signal which amplified and detected according to the application requirement.[14]

2.1.2 Types of sensors

Mainly there are two types of sensors**: Active Sensors and Passive Sensors**

ACTIVE SENSORS

Active sensors require the application of external power for their operation.
This *excitation signal* is modified by the sensor to produce an output signal
Some sensors also called semi-active sensors which use an excitation signal generated by (or radiated from) a source that is not coincident with the sensor.

Active sensors are restricted to frequencies that can be generated and radiated fairly easily. This excludes part of the far infrared (above 3x1012Hz), parts of the ultraviolet band and the gamma

University of He

ray region. These frequency ranges of sensors are used for high commercial applications like Satellite communication.[14]

PASSIVE SENSORS

Passive sensors directly generate an electric signal in response to a stimulus.
That is, the input stimulus is converted by the sensor into output energy without the need for an additional power source. These are mainly rely on a locally generated or natural source of radiation (light from the sun) or an available energy field (gravity).

Passive sensors can exploit EM radiation of any frequency in which some natural phenomenon radiates. This can extend from ELF (**Extreme low frequency**) (below 3x103Hz) up to gamma rays (above 3x1019Hz).

They can exploit acoustic energy (vibration) from infrasound frequencies less than1Hz from earthquakes or explosions up to the ultrasound.[14]

2.1.3 Sensor Output Values

An important part of building a robot is the incorporation of sensors. Sensors translate between the physical world and the abstract world of Microcontrollers. Sensors help translate physical world attributes into values that the computer on a robot can use. The translation produces some sort of output value that the Microcontroller can use. The output of a self-contained sensor or of the remote amplifier of a component sensing system is either digital or analogue

1. Digital Sensors

A digital output is more commonly called a switched output. A switched output has only two states: “on” and “off”. “On” and “off” commonly refer to the status of the load that the sensor output is controlling. The load might be an indicator light, an audible alarm, a clutch or brake mechanism, a solenoid valve or actuator, or a switching relay. The load might also be the input circuit to a timer, counter, programmable logic controller, or computer.

2. Analogue Sensors

University of Hertfordshire

An analogue output is one that varies over a range of voltage (or current) and is proportional to some sensing parameter. For example, the output of an analogue photoelectric sensor is proportional to the strength of the received light signal.

Sensors with analogue outputs are useful in many process control applications where it is necessary to monitor an object's position or size or translucency, and to provide a continuously variable control signal for another analogue device, like a motor speed control.

2.1.4 Evaluating a sensor

Since there are even more types of sensors than physical variables to measure, the user must make a selection based on one or more of these criteria which are very important for choosing relevant sensor for any typical application:

Sensitivity: This is a measure of how much the sensor output changes in response to a given change in the quantity being measured.

Resolution / Minimum detectable measurand: - When deployed, any sensor will respond not only to the measurand, but also to variations in other environmental variables and to inevitable electrical noise. For example, the output of a pressure sensor may also depend on temperature to some degree. The minimum detectable measurand is the lowest measurand level that can be detected in the presence of all other environmental disturbances. This minimum depends on the sensor design, the sensor environment, and the signal processing scheme used to decode the sensor's time-varying electrical output. If the minimum detectable measurand is low, then the resolution is high.

Bandwidth

A sensor with high bandwidth can change its output to follow fast (high-frequency) variations in the input. For measurands that change slowly, such as the temperature of a large liquid tank, bandwidth may not be important. For other applications such as scanning shape or vibration sensors, bandwidth may be the most important figure of merit.

Response Time

Every sensor is specified for its response time. The response time of a sensor or sensing system is the maximum amount of time required to respond to a change in the input signal (e.g. a sensing event). It is the time between the leading edge (and trailing edge) of a sensing event and

University of Hertfordshire

the change in the sensor's output. With a switched output, the response time is the time required for the output to switch from "off" to "on" or from "on" to "off". These two times are not always equal. With an analogue output, the response time is the maximum time required for the output to swing from minimum to maximum or from maximum to minimum. Again, these two times are not necessarily equal.

Response time is important when detecting high-speed events, and becomes quite critical when detecting small objects moving at high speed. Narrow gaps between objects or short times between sensing events must also be considered when verifying that a sensor's response is fast enough for the application.[3]

Static characteristics:

These describe the device performance at room conditions with very slow changes in the measurand and in the absence of any mechanical shock. Static characteristics include[3]:

Linearity:

A sensor is said to be linear when the constant of proportionality has the same value with in the whole measurand range and then exhibits increasing deviation from straight line linearity.

Accuracy

It is the capacity of a measuring instrument to give RESULTS close to the TRUE VALUE of the measured quantity.

Precision

The capacity of a measuring instrument to give the same reading when repetitively measuring the same quantity under the same prescribed conditions Precision implies agreement between successive readings, NOT closeness to the true value. Precision is related to the variance of a set of measurements. Precision is a necessary but not sufficient condition for accuracy

Two terms closely related to precision

Repeatability

The precision of a set of measurements taken over a short time interval

Reproducibility

University of Hertfordshire

The precision of a set of measurements but taken over a long time interval or

Performed by different operators or with different instruments or in different laboratories

Input range

The maximum and minimum value of the physical variable that can be measured (i.e., -40F/100F in a thermometer)

Output range can be defined similarly.

Linearity

The closeness of the calibration curve to a specified straight line (i.e., theoretical behavior, least-squares fit)

Monotonicity

A monotonic curve is one in which the dependent variable always increases or decreases as the independent variable increases.

Hystheresis

The difference between two output values that correspond to the same input depending on the trajectory followed by the sensor (i.e., magnetization in ferromagnetic materials) Backslash: hystheresis caused by looseness in a mechanical joint

Dynamic Characteristics

The sensor response to a variable input is different from that Exhibited when the input signals are constant (the latter is described by the static characteristics)

The reason for dynamic characteristics is the presence of energy-storing elements

Inertial: masses, inductances

Capacitances: electrical, thermal

Dynamic characteristics are determined by analyzing the response of the sensor to a family of variable input waveforms: Impulse, step, ramp, sinusoidal, white noise.

University of Hertfordshire

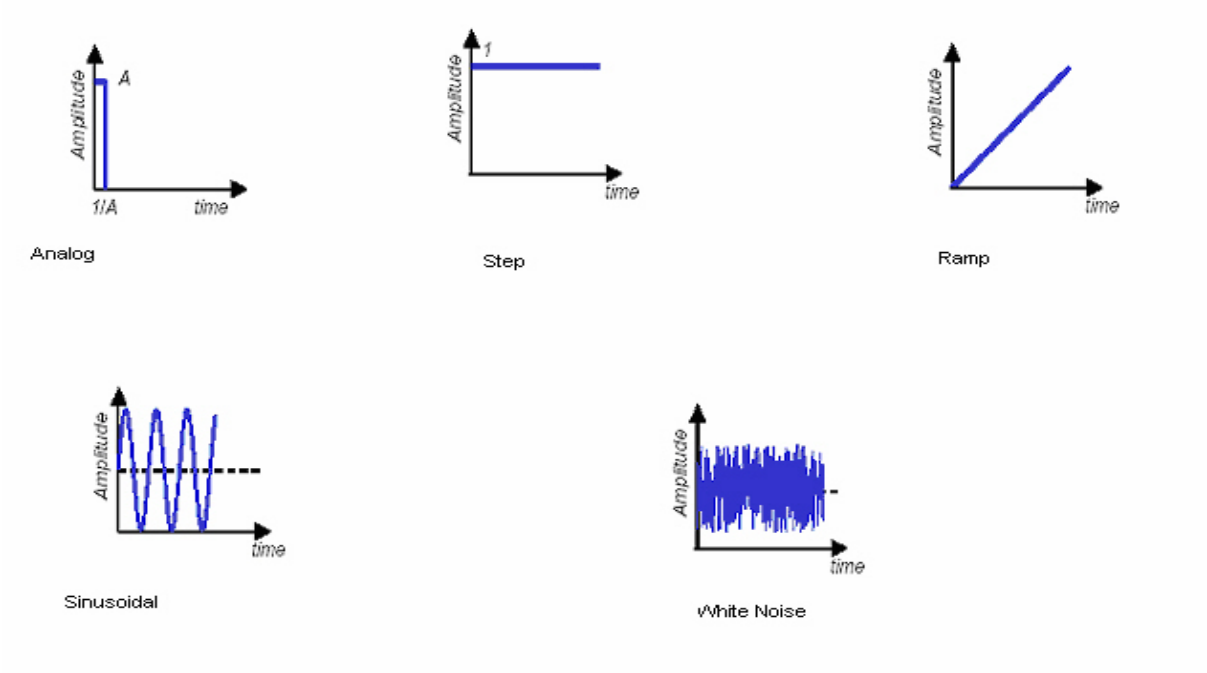


Fig 2.4 Different Types of Signals

2.1.5 Advantages and Disadvantages of the sensors

Advantages

Passive	Active
* Low power requirement (Long life) * Simple principles (Sometimes) *Cannot be detected (Covert) *Large variety	* Long range operation possible * Measurement requirements matched to transmitter Characteristics *High range and angle resolution possible *Radiation pattern Constrains Observations

Disadvantages

Passive	Active
*Target of interest must radiate or modify the field (elec., gravity). *Availability not guaranteed (no light, contrast etc.). *Typically short range though not always(astronomy) . *Prone to feature ambiguity and error of scale (e.g. Angle only).	*Large power requirement. *Easy to detect (not covert). * Complex transmitter and receiver logic. determined *Reliability by transmitter and receiver only.

2.2 Lego Mindstorms and DCP sensors

There is a similar project called " *The Lego Mindstorms Robots* " from which author take a basic idea for this project.Initially author decided to use the sensors of this system in his project as described in the project background. The Information on the light sensor, touch sensor and distance sensor input values were found on "MindStorms RCX Sensor Input Page" website .Various experiments had conducted to make these sensors appropriate for using in this project but found that the inputs and outputs of these sensors were taken from the same wire .For example in light sensor the input sends a high (8 VDC) level for around 1.2ms and disconnects it for around 0.1ms which is when it reads the sensor as output. A brief description of these sensors is given here for the reader or designer who wants to work anymore on these Sensors. The circuit shown in figure 2.3 is used to interface these sensors other than Lego robot.[15]

University of Hertfordshire

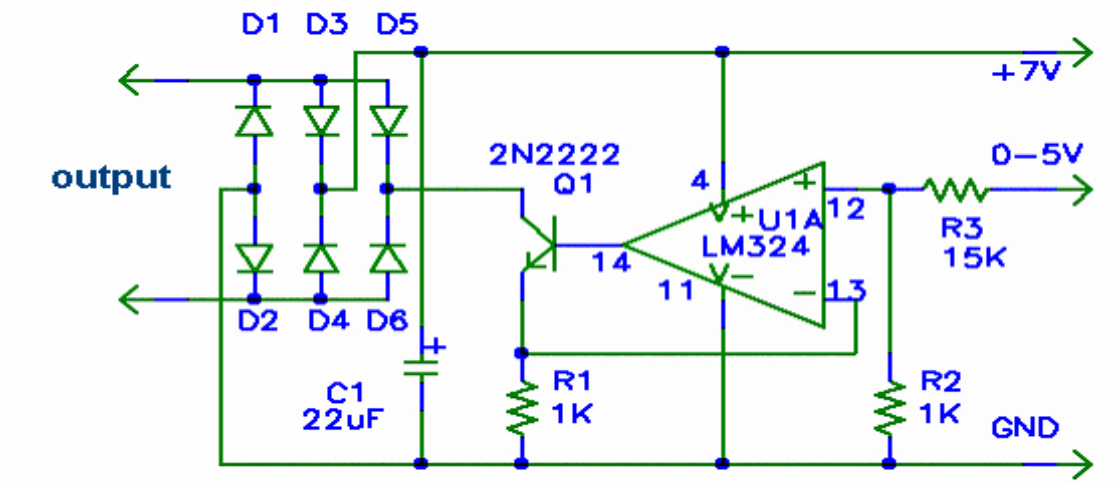


Figure 2.3 General purpose analogue interface circuit

This circuit should work as a more general purpose analog interface. The Op-amp provides gain (1+R2/R3) and could be used where the input signal is too small for the 0 to 5 volt range. The +7V is an auxiliary output from the circuit to provide about 10mA of power to operate other sensor electronics. NPN transistor is added to get more linear interface.

Touch Sensor

For a touch sensor input, the interface circuit has a 10,000 ohm resistor pulling up the input to 5V. The voltage on the input is converted to an internal RAW value in the range 0V = 0 to 5V = 1023. Depending on the sensor type, the RAW number is converted into the number we see in the program, Test Panel or View. In the case of the touch sensor it is converted into a Boolean 1 or 0 values.

Light Sensor

For a light sensor the interface circuit uses a 120 ohm resistor to pull up the input to about 8V to power the red LED for about 3ms and then looks at the sensor voltage for a 0.1ms. During the 0.1ms time the sensor is read just like the touch sensor. The light values are calculated: Light = 146 – RAW / 7 within the range of 0 to 100. The table below shows the percent (Pct) values of light with the equivalent Boolean and RAW outputs.

Rotation Sensor

University of Hertfo

The interface for the Rotation sensor looks just like the Light sensor. About 8V is applied for 3ms and then a voltage is read while 5V is applied through a 10,000ohm resistor for 0.1ms. The sensor only outputs 4 analogue values of 1.8v, 2.6v, 3.8v and 5.1v. Each voltage corresponds to 22.5 degrees of rotation so there are 16 counts per rotation. The table 2.1 below shows the percent (Pct) values of rotation as used by the light sensor with the equivalent Boolean and RAW outputs.[4]

Volts	**Raw**	**Sensor ohms**	**Light**	**Touch**
0.0	0	0		1
1.1	225	2816		1
1.6	322	4587	100	1
2.2	450	7840	82	1
2.8	565	12309	65	0
3.8	785	32845	34	0
4.6	945	119620	11	0
5.0	1023	Inf	0	0

Table 2.1 Summary Input Table[4]

DCP Sound Sensor

Sound sensor of DCP series ordered from the Commotion group which is a mail order company specializing in the supply of Science, Technology and Primary School Products for using in this project. DCP also provides Adapter Lead which allows connecting all the DCP sensors with Lego Mindstorms robots as well. So many experiments conducted on this sound sensor to find the function of five pins of this sensor but no information found for its internal circuitry even after contacting support group of DCP micro sensors .So author decided not to take this sensor as input for his system. The brief description of this sensor pin-out socket as viewed from outside is given for reader who wants to continue further research on the DCP sensors [5]

In figure2.4, function of two extra pins also shown for using same socket for all digital and analogue sensors of DCP series.[5].

University of Hertfordshire

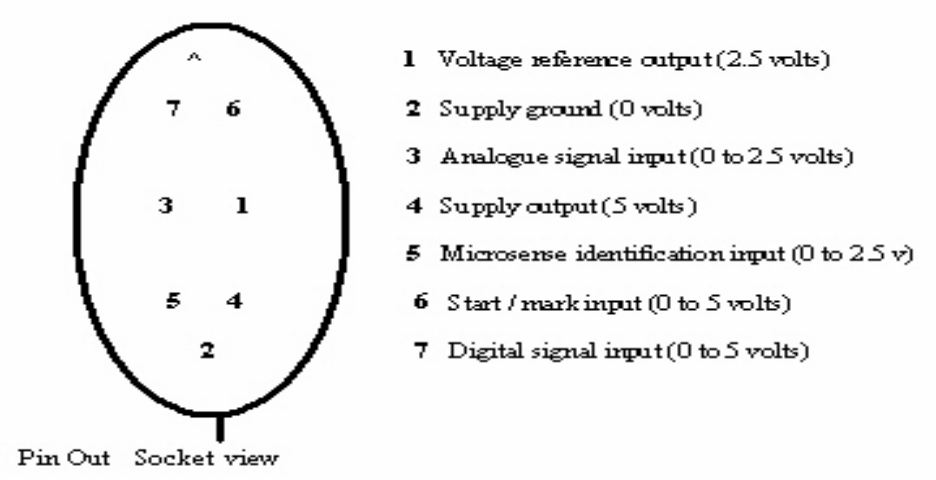

7 6
3 1
5 4
2
Pin Out Socket view
1 Voltage reference output (2.5 volts)
2 Supply ground (0 volts)
3 Analogue signal input (0 to 2.5 volts)
4 Supply output (5 volts)
5 Microsense identification input (0 to 2.5 v)
6 Start / mark input (0 to 5 volts)
7 Digital signal input (0 to 5 volts)

Figure 2.4 Function of pins of DCP sensors

2.3 Choice of the sensors

After the study of above discussed Light sensor, Touch sensor, Rotation sensor and Sound sensor , it is concluded that these sensors were designed by their companies according to their own software's and design considerations which they recommend for their own systems. To use these sensors with this project system become more complex and difficult so author decided to design his own simple sensors which were using as input in the flexible development platform of robot.

The detailed design and description of these sensors is given in chapter 5.

2.5 Choice of the Motors

There are mainly two types of motor used in controlling of a robot, the servo motor and stepper motor. Both types of motor have used in same applications in robot like positioning, but they differ in a number of ways.

Stepper motor can be viewed as electric motor without commentators; they are controlled by one step per one pulse. If the load of the motor is not too great, open-loop control is usually used to control the motor. Stepper motors are used in disk drive head positioning, plotters, and numerous other applications.

But for servomotor, they require analogue feedback control systems of some type. Typically, this involves a potentiometer to provide feedback about the rotor position, and some mix of

University of Hertfordshire

circuitry to drive a current through the motor inversely proportional to the difference between the desired position and the current position .Servo motors are used in radio control airplanes to control the position of wing flaps and similar devices

Stepper motor is chosen for this project because of the availability of stepper motor control circuit in the laboratory of University of Hertfordshire designed by research students which saves time to concentrate on other aspects of the project.

But its advantages and disadvantages are also considered before using this motor which are discussed below:

.

Advantages

1. The rotation angle of the motor is proportional to the input pulse.
2. The motor has full torque at standstill (if the windings are energized)
3. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next.
4. Excellent response to starting/stopping/reversing.
5. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependant on the life of the bearing.
6. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
7. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
8. A wide range of rotational speed can be realized as the speed is proportional to the frequency of the input pulses.

Disadvantages

1. Resonances can occur if not properly controlled.
2. Not easy to operate at extremely high speeds.

2.5.1 Control of motor

Controlling of the stepper motors is done by using pulse controlling. The control pulse is positive going pulse with length of 1 to 2 ms which is repeated about 50-60 times a second. The details are in the figure 2.5below:

University of Hertfordshire

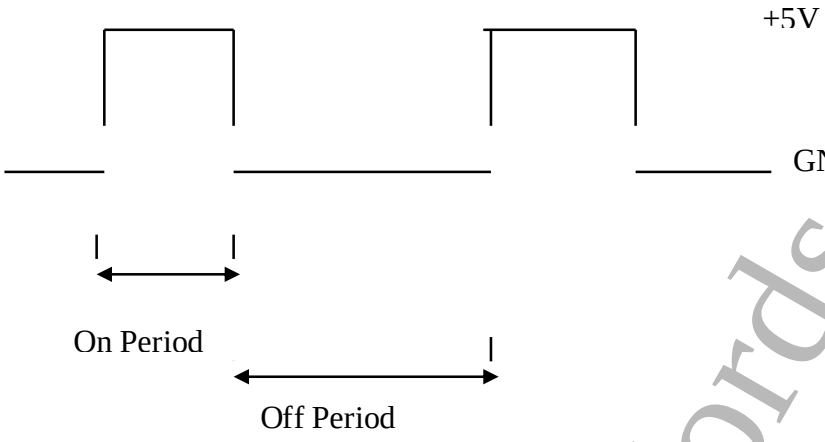


Figure 2.5 – Square Waveform

Stepper motors are controlled by sending them a "pulse" of variable width. The parameters for this pulse are that it has a minimum width, a maximum width, and a repetition rate. These values are not "standard" but there are conventions that are generally accepted. The speed of the motor depends on the duty cycle of the pulse i.e.

Duty cycle = On period +Off period

2.4 Summary

This chapter is for only those readers who are not aware of the basic fundamentals of the sensors and actuators (motors).It consists of the sensor theory with its types and working. Advantages and Disadvantages of sensors are also discussed. Details of Light sensor, Touch sensor and Rotation sensor used in Lego Mindstorms and DCP Logit Sound sensor is given for the reader to work further on these sensors. Finally the types of servo and stepper motor used in the robots with its advantages and disadvantages is described.

University of Hertfordshire

CHAPTER 3

RESEARCH

3.1 Literature search

For the literature search of this project, online databases INSPEC, IEEE and relevant journal articles were searched. Various journals available in Learning Resource Centre of University of Hertfordshire were also referred. Motorola' web site was searched for datasheets relating to the 68000 microprocessor and application notes that were relevant to the project including Program/Interface Timer 68230 and memory mapping techniques. General searches were also conducted on the World Wide Web for any relevant information. Prior to the starting of this project significant research had been undertaken to find out any previous work done on this topic. The results of this research provided the foundation upon which this project is built.

The basic idea was taken from Lego Mindstorm robot which is commercially available for toys but this robot is strictly used for only three sensors and one motor.

3.2 Hardware research

During Literature search no relevant information was found on online database INSPEC about the sensors and motors available in the market. Also search was conducted on the internet to see

University of Hertfordshire

if any similar products were already in production like Lego Mindstorm robot and DCP sensors or any other manufacturer.

The only results found were for Lego Mindstorm systems of which no design details were given, as might be expected for commercial reasons. No information was found for the internal circuitry of the different sensors which were used by current manufacturer because all these sensors were design according to their own software's. A deep research was carried out on Logit Sound sensors (provided by DCP manufacturers), Light sensor, Distance sensor and Touch sensor used in Lego Mindstorm robot.

A research was also conducted for studying advantages and disadvantages of using memory mapping and program interface/timer for interfacing of designed sensors system with 68000.Different analogue to digital conversion (ADC) and digital to analogue conversion (DAC) techniques were also studied.

Detailed hardware design and implementation of information gained from the literature search starts in chapter.

3.3 Software research

For the programming of MT68000 , research was carried out into the relation between 'C' language and microprocessor 68000.Deep study of 'C' language is done for software programming . Research was conducted into specific areas of 'C' language such as various library functions including in microprocessor , features of the Embedded Development Studio(EDS),'C' function call implemented on microprocessor 68000 and using pointers for direct access to a physical address space. Some research was also conducted on the Debugger and Compiler features and various functions of EDS like step over, trace options and implementation of breakpoint in 68000 programs. . Many of the features of the EDS are studied with the online help system present in it.

More detail into the 'C' language software requirements for the project is given in chapter

3.4 Summary

This chapter includes the summary of the theoretical research done by the author in the field of hardware and software for his project and the results obtained from this search.

University of Hertfordshire

CHAPTER 4

'C' LANGUAGE AND THE 68000

4.1 Motorola 68000

The 68000 microprocessor is manufactured by Motorola which contains a 16-bit wide data bus and an internal address bus of 24-bits.It is capable of accessing a linear address space of 16 Megabytes. It is driven by a 10 MHz clock generated by a CMOS oscillator. The 68000 has sixteen 32-bit registers, split into data and address registers. One address register is reserved for the Stack Pointer. Any register, of either type, can be used for any function except direct addressing. Only address registers can be used as the source of an address, but data registers can provide the offset from an address.

Externally the address bus does not include the least significant bit, A0, which means that odd addresses are not addressable. Instead of A0 two data strobes are provided, UDS (Upper Data Strobe) and LDS (lower Data Strobe). These data strobes are active low signals and indicate whether the upper or lower byte of a data word is being addressed. If an entire word is being addressed then both data strobes are asserted. [12]

UDS and LDS can be used within a systems address decoding. At the simplest level UDS and LDS can be connected directly to the chip select (or chip enable) input of two 8-bit wide (data bus) memory devices, thus the upper and lower bytes of the word are stored in physically separate devices.

University of Hertfordshire

One signal of importance to bus transactions is the read/write output (*WR*/). When this output goes low it indicates that the processor has started a write cycle. When the output is high is means that a read cycle is in progress.

The important feature of 68000 is asynchronous data transfers which differentiate from other processors. Handshake signal generated by the device being accessed Indicates that the contents of the data bus is valid .This handshaking signal is used as an input to the 68000 to inform it when an external memory device or peripheral has provided or responded to data on the bus. This handshaking signal is called Data Transfer Acknowledge (DTACK*) and is active low.

If DTACK* is not asserted, CPU generates wait-states until DTACK goes low or until an error state is declared.

When DTACK* is asserted, CPU completes the current access and begins the next cycle DTACK* has to be generated a certain time after the beginning of a valid memory access (timer supplied by the system designer)

The asynchronous bus therefore allows the 68000 to be used with slower memory devices and peripherals than would otherwise be possible.

As in the 68000, synchronous bus control group of signals (VPA*, VMA*, E) are not needed because all data transfer may take place asynchronously. But the synchronous bus control group simplifies the interface between the 68000 and peripherals designed for use with the older 8-bit synchronous bus microprocessors like 6800.For synchronous data transfer three control signals Valid peripheral address(VPA*),Valid memory address(VMA*) and Enable(E).

VPA*-This indicates that a synchronous bus cycle is requested by the external peripheral.. When the processor recognizes that VPA*has been asserted, it initiates a synchronous data transfer by means of its VMA* and E control signals.

VMA*-The active low valid memory address signal from the 680informs the peripherals that there is a valid address on the address bus. The assertion of VMA* by the CPU is a response to the assertion of VPA* by an addressed peripherals.

E-The enable output from the 68000 is a timing signal derived from the 68000's own clock input. One clock cycle is equal to ten 68000 clock cycles and it is non-symmetric that is low for six clock cycles and high for four.

University of Hertfordshire

With a synchronous bus, memory devices would have to be fast enough to have data ready when the processor expects it and peripherals fast enough to act on data before the processor removes it from the bus.

There are three signals called function codes (FC0, FC1 and FC2) that are output from the 68000. The functions codes provide a means of determining what state the 68000 is currently in as shown in Figure 4.1..Three of the eight states are marked undefined ,reserved which tells us that these states may be reassigned in the future versions of the 68000. The function codes and the current address on the bus are considered valid when the address strobe (AS) is asserted active low by the processor

Function Code			
FC0	**FC1**	**FC2**	**Meaning**
0	0	0	Reserved
0	0	1	User data being accessed
0	1	0	User program being accessed
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Supervisor data being accessed
1	1	0	Supervisor program being accessed
1	1	1	Interrupt being processed

Figure 4.1 68000 function codes

The 68000 has 64 pins arranged in nine groups as shown in Figure4.2 . Each of these groups is labeled by the function of pins it performs.

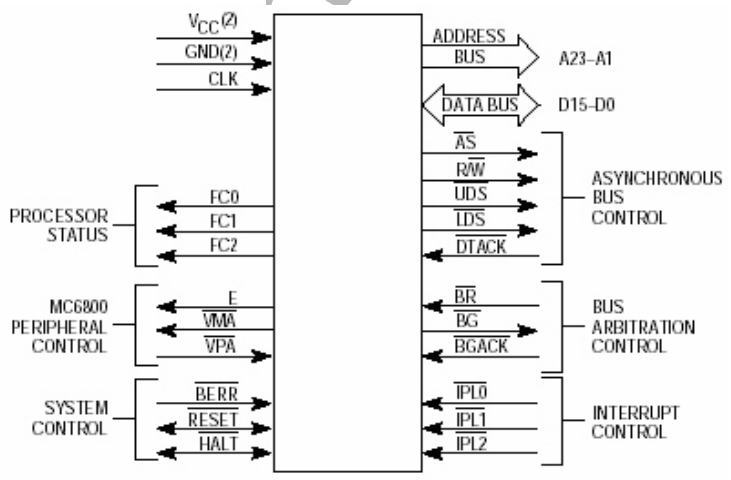


Figure 4.2 Logical arrangements of the 68000 pins[6]

4.2 FLIGHT-68K Board

The FLIGHT-68K is designed by the Flight Electronics to provide the educational training to the students or programmer in the world of 16/32 bit microprocessors by the way of the very popular Motorola MC68000. The heart of the Flight FLT-68K is the MC68000 CPU and it has a 16-bit data bus and 24-bit address bus. The system also incorporates a range of peripheral devices and is driven by a 10MHz clock. The block diagram of the FLIGHT 68K is shown in Appendix A.1.

The FLT-68K is supplied with 16K bytes of RAM, which is more than sufficient for most teaching requirements, but is expandable to a total of 512K bytes making it also suitable for larger scale applications. Two RAM memory sockets accept 6264 8K, 62256 32K, or 551001 128K static RAM in pairs resulting in 16K, 64K and 256K bytes respectively. Two further auxiliary RAM sockets may also be used to accommodate these same devices resulting in a maximum RAM size of 512K bytes. The most common configuration is by connecting the serial port of a personal computer to the F68k. Figure 4.3 shows the setup between the F68k and a personal computer running a serial port terminal..

68000

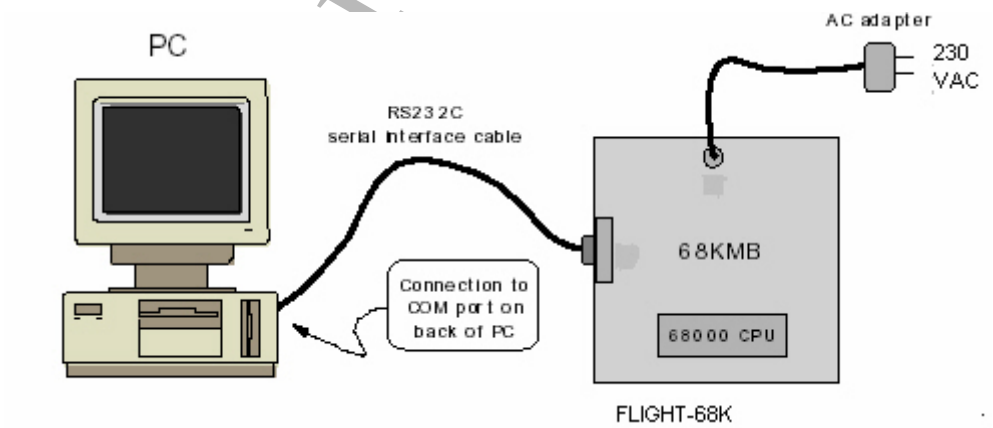


Figure 4.3: Connection of PC and FLIGHT

Also as standard, two 27256 32K EPROM are fitted containing the FLT-68K's monitor firmware. This firmware comprises comprehensive monitor, line by line assembler and

dissembler. The monitor program provided with the FLT-68K has 53 different commands which is invoked by a simple two letter code .It is user-friendly and easy to use .The total descriptions of these commands are available in FLIGHT 68K user manual. The line-by-line assembler accepts 68000 assembler language instructions and assembles them a line at time, making program entry simple and quick. Using the disassembler, **(**M68000 8-/16-/32-BIT MICROPROCESSORS USER'S MANUAL MOTOROLA (3.1) may be disassembled from anywhere in memory. The assembler source code may be displayed, or printed out if a printer is connected.

There is also a C compiler available for the F68k. The compiler replaces the monitor EPROM on the F68k microprocessor board and the terminal window on the personal computer is replaced with a program called Embedded Development Studio (EDS).

The Flight-68K includes 68681 Dual UART and 68230 Parallel Interface/Timer enabling to learn about the 68K family peripherals and to apply the FLIGHT-68K to real control applications.[7].

The MC68230 provides 24 parallel input/output lines plus a counter/timer. Access to these lines is by the 40 way I/O connector which also links to the Switch/Lamp Unit, 4mm I/O Module, or to our Application Board which provides a wide range of ready built control applications There are two ports on the PI/T used as inputs and outputs which are mapped into the memory map starting at 800001. The basic addresses of these ports are given in table 4.1.

PI/T	Port A address	Port B address
Data Direction Register	800005	800007
Control Register	80000D	80000F
Data Register	800011	800013

Table 4.1 MC68230 PI/T addresses of F68k

Further user expansion may be gained by way of the male 64 way DIN41612 connector, which provides access to the entire MC68000processor signal

University of Hertfordshire

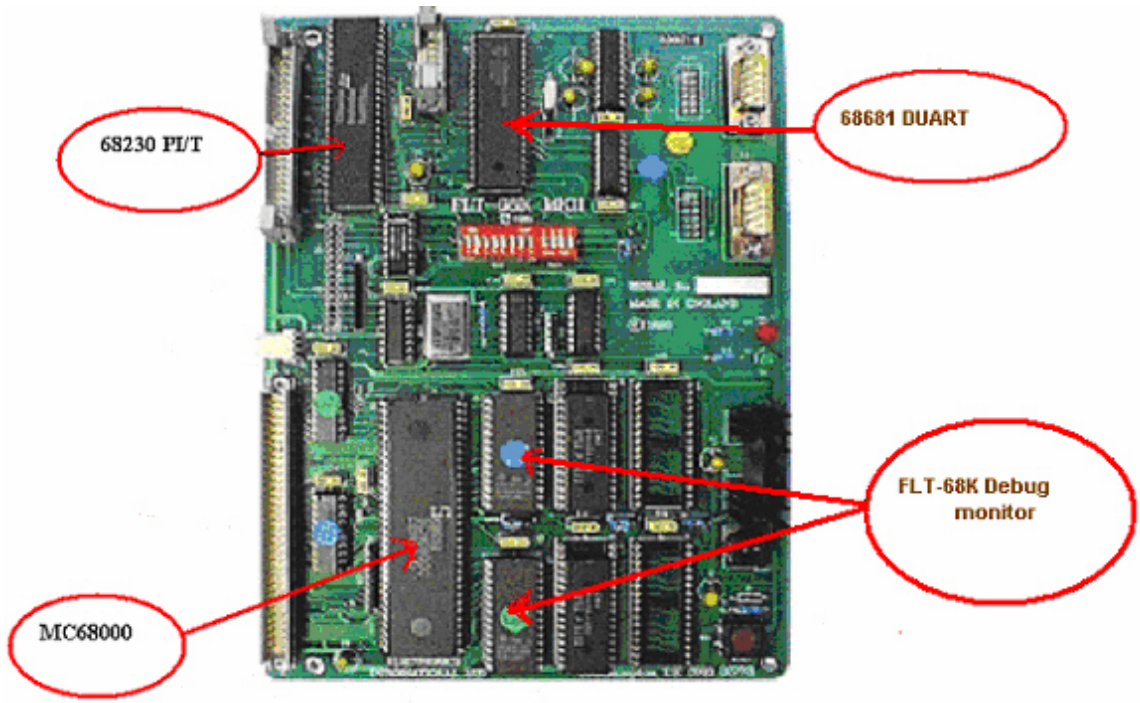


Figure 4.4: FIGHT 68K

An on-board system reset switch is provided, as are status LED's, and fault switches which allow the simulation of hardware faults for student fault-finding exercises for practice. 16-bit microprocessor trainer designed to introduce students to the industry standard microprocessor for industrial control applications, the Motorola 68000 CPU.

4.3 Choosing 'C' language for programming

In times past, computer languages were classified into : low-level language (assembler), and high-level languages (e.g. COBOL, FORTRAN, ALGOL). Assembler is related to the actual machine instructions, that is, it depends on the actual hardware. One statement in high-level language will expand into many, many assembler statements, and is independent of the hardware. [8]

Assembler language is not standardized. There is one assembler language for Z80 micro processor, another for 68000, another for 6502 microprocessor etc.

C may be regarded as "standardized assembler", and shakes off its dependence on particular microprocessor. C may be regarded as evolved from "Assembler -->> B (written by Ken

University of He

Thompson in 1970 for PDP-7) ->> BCPL (written by Martin Richards) ->>C (written by Dennis M. Ritche)".[8].

Nowadays, people rarely use "assembler" any more. Nearly all the chip manufacturers provide a "C" compiler for the chip they manufactured. The C language was chosen to program the MC68000 as it is one of the most popular and powerful high level language. Consideration was given to assembly language because it is more efficient in terms of code but is more complex for programming.

C had many advantages over assembly language for this application:-

- Specific knowledge of the microprocessor instruction set is not required ,a basic knowledge of the MC68000 memory structure is desirable but not essential.
- Programming and test time is reduced, therefore increasing efficiency.
- Programs take a formal structure and can be split into separate functions.
- Details of register allocation and addressing of the various data types is managed by the compiler.
- The ability to combine variable names with specific functions helps to improve readability and program construction.
- C libraries contain many standard routines.
- It can be compiled on a variety of computers.
- The C language is very portable; code can be quickly adapted for another processor for future development.

3.4 Summary

This chapter highlights the reasons for choosing MC68000, FLIGHT- 68K training board and C language for programming with the brief description of the 68000 microprocessor and Program Interface/Timer MC68230.

CHAPTER 5

HARDWARE DESIGN

University of Hertfordshire

5.1 Design and testing of the Sensors

The purpose of the sensor subsystem is to enable the robot to have some appreciation of its physical surroundings. A broad range of sensors is available for practical use but it is found that most of the sensors designed by Robotics companies are designed according to their software specifications .For example DCP sound sensor has Micro sense identification input at pin number 5(see section 2.2) to differentiate from other sensors in their software. This signal is used to identify sensors.

So for this project decision is made to design simple sensors using suitable components with operational amplifier 741. An operational amplifier IC is a solid-state integrated circuit that uses external feedback to control its functions. It is one of the most versatile devices in all of electronics. The data sheet and pin specification of Op Amp 741 is given in Appendix 6.This Op Amp has so many Linear and Non-linear applications such as differential amplifiers, log amplifiers, comparators, and oscillators, precision rectifier etc.

The designs of sensors described below are using simple application of Op Amp which Voltage Amplifier .The Voltage amplifier is shown in figure. 5.1

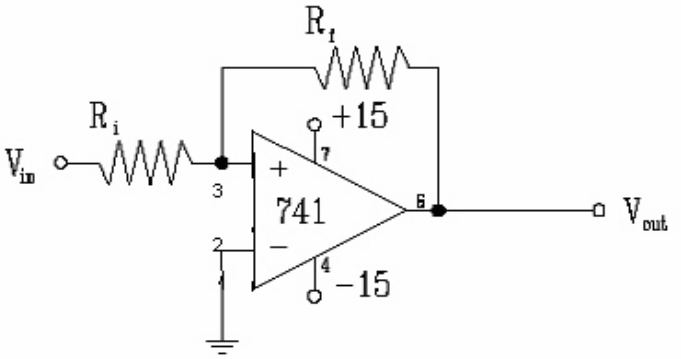


Figure 5.1 Voltage Amplifier Using Op-Amp 741

Its ideal transfer function (ideal gain) equation is given by

Vout / Vin = (Rf + Ri) / Ri

While it's actual gain equation is given by the control function relationship as

Vo / Vin = A (Rf+Ri) / 1+A [Ri/(Rf+Ri)]

Univ of Hertfordshire

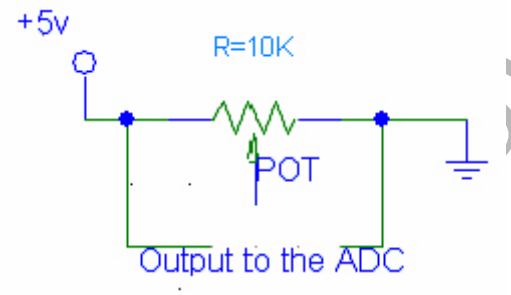
A is the gain of the Op Amp. (Refer Appendix).

Design and testing of the simple Voltage Sensor, Switch Sensor, Light Sensor and Sound Sensor is given below.

5.1 Voltage Sensor

Potentiometers are especially used for making angular measurement devices because most pots only turn approximately 270 degrees or so. They are great for determining the angles of a robot arm. Actually potentiometer converts a mechanical input into electrical signal.

Potentiometer is used to design this voltage sensor. In figure 5.2 resistance of a sensor is converted into a voltage by using potentiometer, which is almost linear as shown in graph (figure 5.3)



Simple Voltage Sensor Circuit

Figure 5.2

University of Hertfordshire

Pot Resistance (in K Ohms)	Output Voltage (in Volts)
10.2	**5.1**
9.4	**4.7**
8.4	**4.2**
7.3	**3.6**
5.9	**3.0**
4.8	**2.4**
4.1	**2.1**
2.6	**1.3**
1.2	**0.7**

Table 5.1 Output of the Voltage Sensor

The input supply of reference voltage sensor is tried to the position of the resistance of changes and voltage changes. Output Voltage +5v is given as and output of this taken according to pot. As it varies, the potentiometer accordingly output Resistance Vs graph is plotted in figure.5.3. Potentiometer (pot) resistance is noted according to the position of the pot.

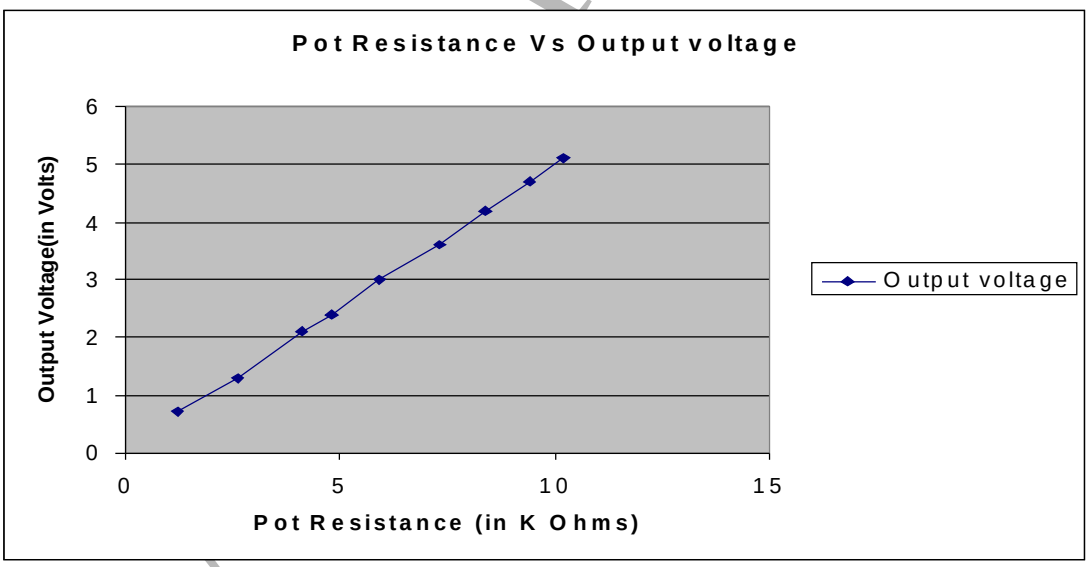


Figure 5.3: Graphical representation of output of Voltage sensor

5.1.3 Switch Sensor

One of the most basic of all sensors is a simple switch. Switches are used in bumper sensors, to detect limits of motion, for user input, and a whole host of other things.

University of Hertfordshire

Switches come in two types: normally open (NO) and normally close (NC). Many micro switch designs actually have one common terminal, and both a NO terminal and a NC terminal.

The circuit in figure 5.4 shows how to potentiometer is used as a analogue switch sensor Reference voltage of 2.1 V is given at the inverting terminal of Op Amp by a Zener Diode .This voltage across Zener Diode switch the output voltage between 2.1 V and 4.9 V by rotating the knob of potentiometer from minimum to higher position. The Op Amp is used in non-inverting mode which acts as amplifier only.

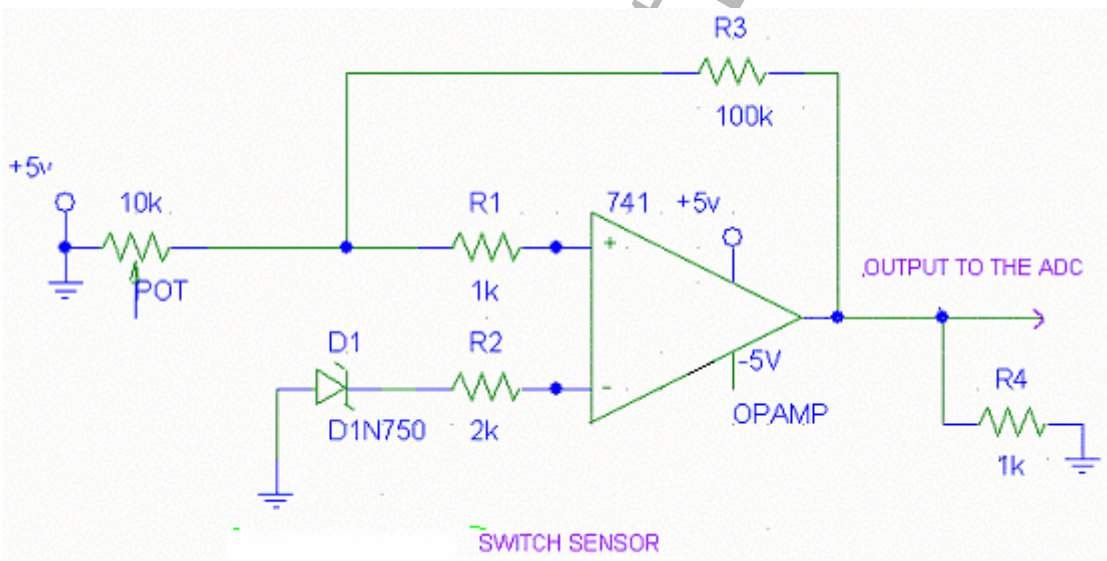


Fig 5.4 Switch Sensor

5.3.3 Light Sensor

A simple LDR application is implemented for designing light sensor. When there is NO light falling on the sensor, the resistance of the LDR becomes infinite. When light falls on the LDR, the resistance changes according to the intensity of light. Example uses for this circuit: Dark-room, Automatic door-lock, closet, Shed Alarm, The data sheet of LDR is present in Appendix H. Since there is no light meter to measure the light intensity therefore graph is plotted between

the resistance of the LDR and the Output Voltage by hit and trail method. This resistance may not be exactly match with the actual light intensity. The plot is shown in figure 5.6.

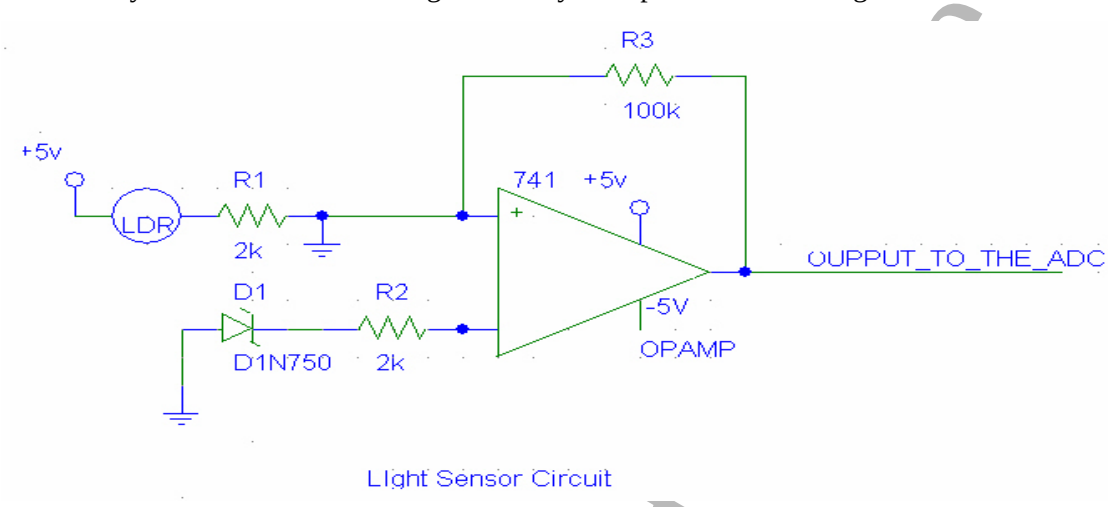


Figure 5.5

LDR Resistance (in KOhms)	Output Voltage (In Volts)
27	0.5
22.1	0.9
20.2	1.2
15.3	1.6
10.2	1.9
1.92	2.1
1.71	2.4
1.6	2.6
1.04	2.84

Table 5.1 LDR Resistance Vs Output Voltage Of light Sensor

University of Haiti

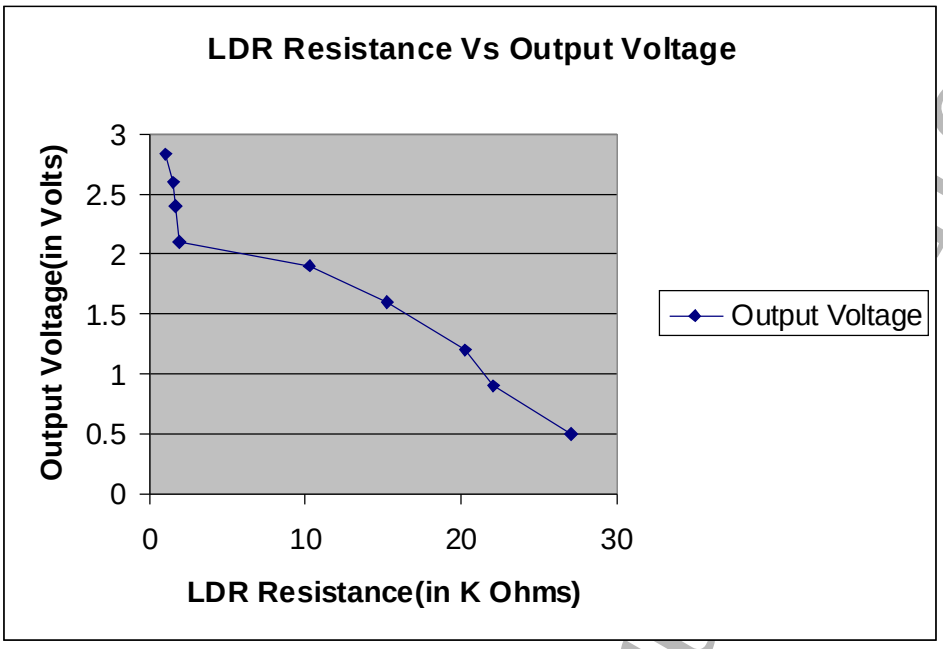


Figure 5.6: Graphical representation of the light sensor output

5.1.4 Sound Sensor

The circuit diagram for simple sound sensor is shown in figure 5.8 in which the Op Amp is used only to amplify the output signal of microphone so that it easily controlled the motor. The lower voltage signal cannot control the speed of the motor as the minimum input supply to the motor is 2.5 V. This circuit works very well only when loud noises are present. For the specifications of the micro phone refer Appendix.

This circuit gives very less output changes because of not proper amplification and also the poor quality micro phone. The output varies in only mill volts due to which changes in the speed of the motor are not notable.

University of Hertfordshire

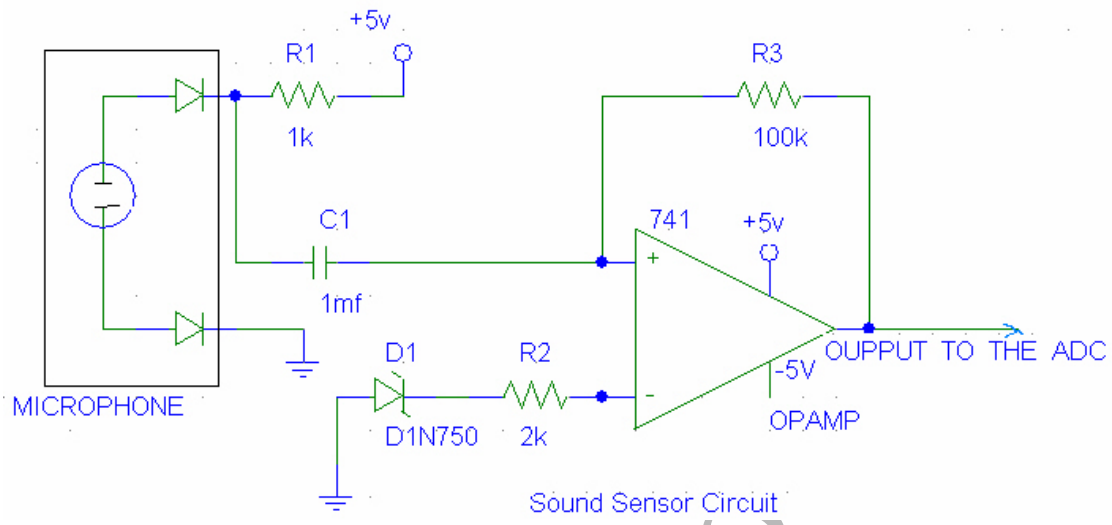


Fig 5.7 Sound Sensor Circuit Diagram

5.2 Analogue to Digital converter

Microprocessors almost always deal with discrete values. Processor such as the 8085 deal with 8 bit values. An important part of using an Analog Signal is being able to convert it to a Discrete Signal such as a 8-bit digital value. This allows the Microprocessor to do things like compute values and perform comparisons. Fortunately, most modern processors have a resource called an Analog to Digital converter (A/D converter).[9]

The function of the A/D converter is to convert an Analogue signal into a digital value. It does this with a mapping function that assigns discrete values to the entire range of voltages. It is typical for the range of an A/D converter to be 0 to +5 volts.

The A/D converter will divide the range of values by the number of discrete combinations. For example, the table 5.2 shows the samples of an Analogue Signal that have been converted into digital values [9]

> Volts	< Volts	Conversion

University of Hertfo

0.0000	0.0195	0
0.0195	0.0391	1
0.0391	0.0586	2
0.0586	0.0781	3
0.0781	0.0977	4

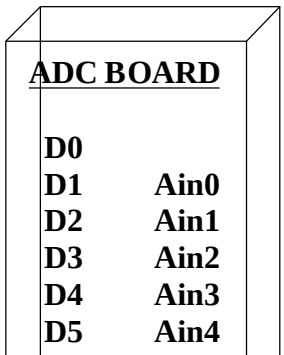
Table 5.2 Analogue Signal Converted into Discrete Values

These digital values are then expressed as a digital number, using a binary numbering system consisting only of 0's and 1's. The resulting digital codes can be used in various communications systems. The ability to convert analogue signals to digital and vice-versa is very important in signal processing. The objective of an A/D converter is to determine the output digital word corresponding to an analogue input signal .Different types of the ADC are used as per the applications. Main types of A-D converters are

- Successive Approximation Register(SAR)
- Dual slope integrating
- Charge balancing
- Flash Converter
- Sigma-delta Converter
- Integrating ADC

Details on these ADC can be finding from any basic Electronics book.

The ADC used in this project is available in microprocessor which is designed by Research Students .The Study and testing (refer chapter 7)of this board is very necessary before using .The basic block diagram of ADC is shown in Figure 5.8 in which



University of Hertfordshire

Figure 5.8: ADC Board

Ain0-Ain are the input pins and D0-D7 are the output pins. CS* is the active low chip select pin which takes signal from VMA (Valid Memory Address) to active the ADC.

To select single input from the eight inputs binary data is given on the A0, A1, A2 pins as shown below.

A0	A1	A2	Input Pin Selected
0	0	0	Ain0
0	0	1	Ain1
0	1	0	Ain2
0	1	1	Ain3
1	0	0	Ain4
1	0	1	Ain5
1	1	0	Ain6
1	1	1	Ain7

Table 5.3: Data for selecting ADC Input Pin

RS-232 a serial communication link between the PC and the 68000 it is a standard that describes a common method of serial signaling using a voltage that switches between a positive level for logic "0" and a negative level for logic "1".

5.3 Bus Expander

University of Hertfordshire

Bus Expander is designed as a prototype board for the connection with MC 68000 chip. This board consists of all the 40 pins of 68000 microprocessor and connected with FLIGHT-68K board via 40 pin connector.

Figure 5.9 is the block diagram of this board which contains only pins of MC6800 used in this project. The VPA (Valid Peripheral Address) and VMA (valid Memory Address) signals are used rather than DTACK (Data Acknowledge) because of not designing any wait state generation circuitry and also the zero wait-sate time is less than the minimum access time for the ADC chip.

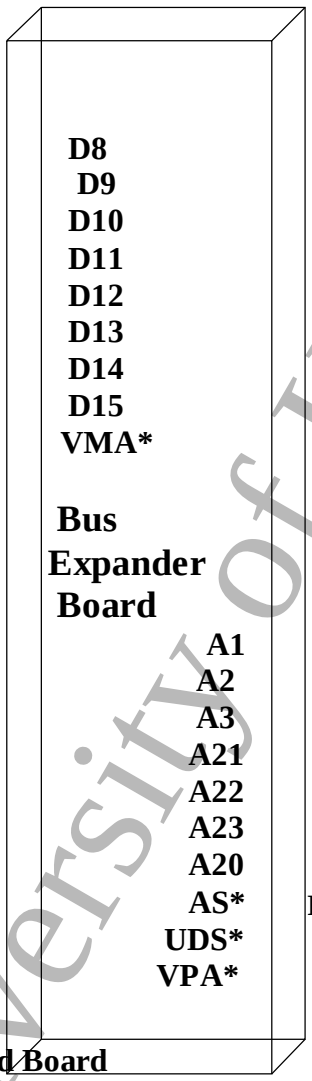


Figure5.9: Bus Expander Board

5.4 I/O Buffered Board

The Buffered IO board is used to interface the MC68230 Peripheral Interface / Timer (PIT) on the Flight 68K main board to external peripherals.

The board has 12 input pins (PA0 – PA7 and PC4-PC7) and 12 output pins (PB0-PB7 and PC0-PC3). The logic value of the pin is displayed on an LED next to it.

The input pins have a choice of signal source. The input can either be set by the value

University of Hertfordshire

0f the signal applied to the 2mm connector, or from the switches on the board. The input source is set with the double switch towards the bottom of the board.
To use the board, connect it to the flight 68k board via the 40 pin IDC connector [13]

Power is supplied to the board via one of the two power connectors to the right of the IDC connector. The power source must be 5 volts and can be sourced from a 5volt PSU, or from daisy-chaining a power cable from another board.
The circuit diagram of I/O buffered board is shown in Appendix E.

Before using the board it must set the ports to the appropriate settings in the program, as the board will not work otherwise. The Port data direction registers is set to the appropriate values, like Port A Data Direction Register needs to be set to input (0x00) and Port B to output (0xff) and Port C to (0x0f).Other registers are to be set depending on the application, as shown in the program in Appendix B.2. Before implementing in the project initial testing is done which is given in Chapter 7.

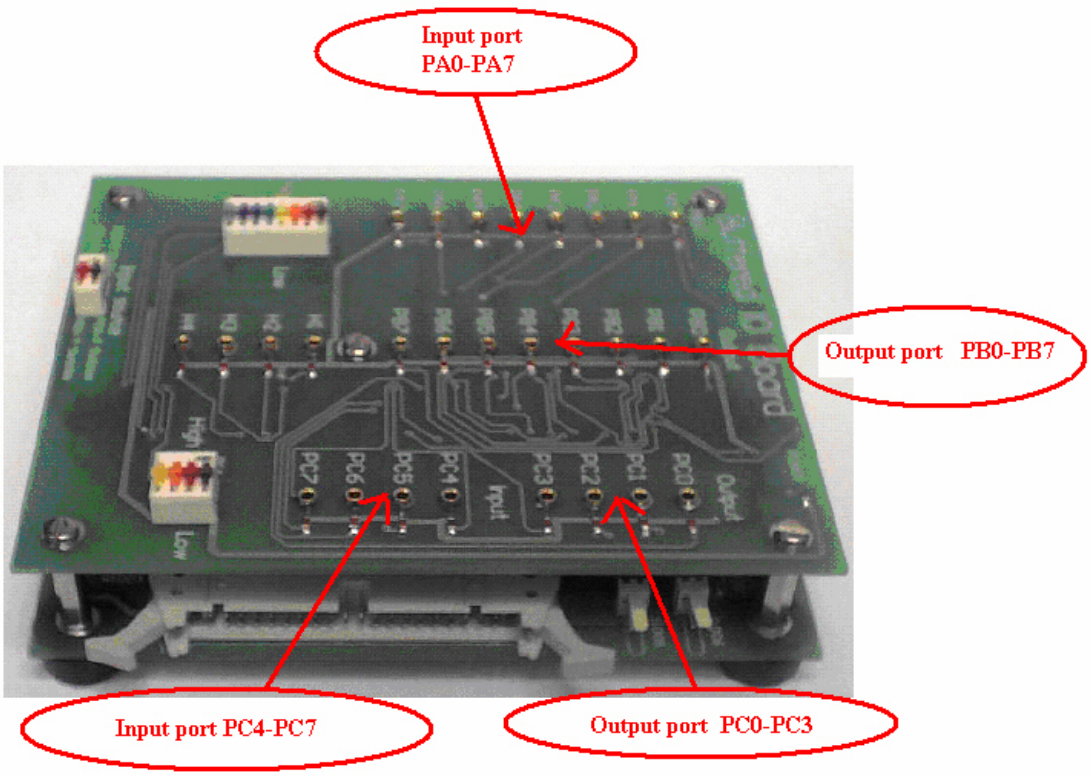


Figure 5.10:- I/O Buffered Board [13]

5.5 Stepper Motor Controller Board

University of Hertfordshire

The stepper motor is used at the output and its speed is controlled according to the physical parameter of the environment sensed by sensors. The stepper motor controller board is taken from Microprocessor lab of University of Hertfordshire which saved the time spent in designing motor controller circuit.

This stepper motor module is a stepper motor with a disc marked with 12 segments and an index marker on the number one. The connections are explained below.[13]

Inputs:

Step: One step movement is generated for each negative edge of the step signal.

Direction: Logic '1' for anticlockwise movement. Logic '0' for clockwise.

Half Step Mode: Logic '1' for full step and logic '0' for half step mode. 4 LED's show the step sequence driving the motor.

Inhibit: Logic '1' to Inhibit driver ic and logic '0' to enable driver ic / module.

Outputs:

Segment pulse: Optical sensors detect each segment / number and produce a logic '1' pulse accordingly.

Index Pulse: Logic '1' output when disc rotates to alignment with number 1 as shown above. LED's indicate the pulse outputs. [13]

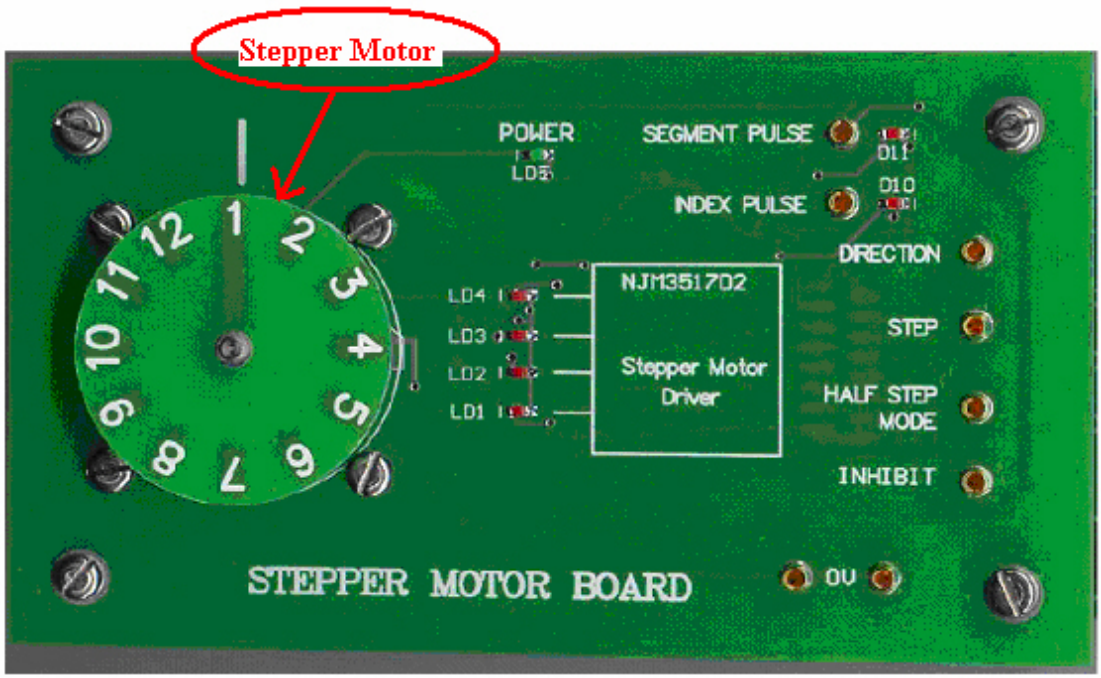


Figure: 5.11 Stepper Motor Controller Board[13]

The circuit diagram of stepper motor board is shown in Appendix.D

University of Hertfordshire

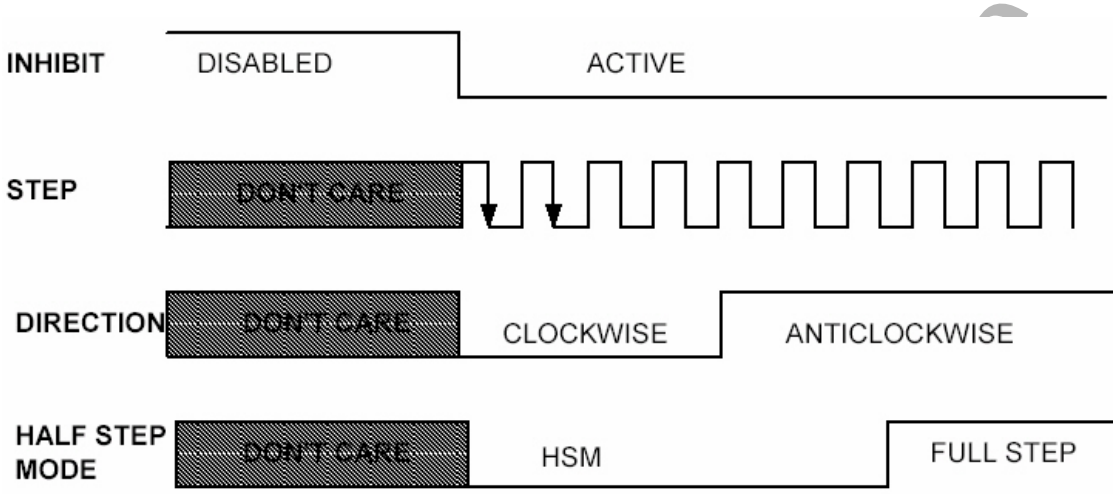


Figure 5.12: Timing Diagram of Different Modes of Stepper Motor[13]

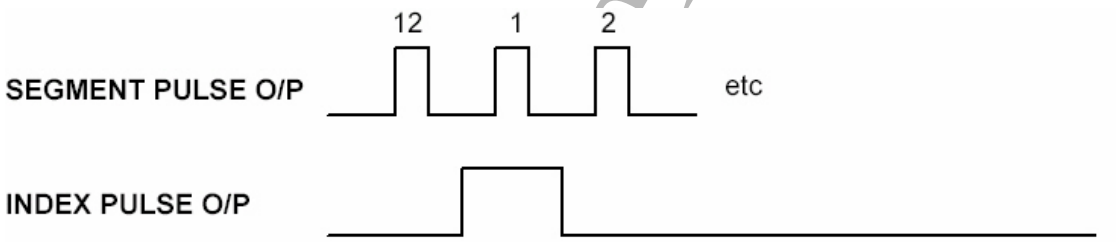


Figure 5.13 .Timing Diagram of Segment and Index pulse[13]

5.6 Address Decoding

Externally, the Motorola 68000 is seen as having a 23-bit address bus as it does not provide address line A0 with the consequence that odd addresses can not be accessed. Instead the 68000 provides an upper and lower data strobe (UDS and LDS) which are used to select an upper or lower byte of the memory word addressed by A01-A23.

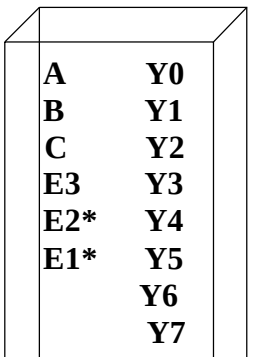
To select one input of ADC is working at a time, full address is chosen. Any Microprocessor is said to have full address decoding when each addressable location with in a memory component respond s only to a single, unique address on the system address bus. In other words, each physical memory location, either by specifying a given device or by specifying a address with in it. [3]Full address decoding can be applied to the problem of distinguish two or more memory components or any external device like any one input of the ADC is selected according to the following address decoding table 5.4

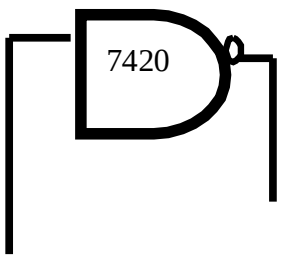
University of Port

A23 A22 A21 A20 A19 A18 ---------- A5 A4 A3A2A1A0	DCPTR address	DC Input Selected
1 1 0 0 0 0 0 0 0 0 0 0 0	D00000	Ain0
1 1 0 0 0 0 0 0 0 0 1 0	D00002	Ain1
1 1 0 0 0 0 0 0 0 1 0 0	D00004	Ain2
1 1 0 0 0 0 0 0 0 1 1 0	D00006	Ain3
1 1 0 0 0 0 0 0 1 0 0 0	D00008	Ain4
1 1 0 0 0 0 0 0 1 0 1 0	D0000A	Ain5
1 1 0 0 0 0 0 0 1 1 0 0	D0000C	Ain6
1 1 0 0 0 0 0 0 1 1 1 0	D0000E	Ain7

Table 5.4 Address Decoding Table

The 74138, 3:8 decoder is used for full address decoding shown in figure 5.14 .Y0-Y7 are the output pins of 3:8 decoder which are selected according to the status of the A,B,C pins .E1*(active low),E2*(active low) and E3(active high) are enable pins which are connected to the UDS*, AS* and A20 pins of MC68000 respectively.





University of Hertfordshire

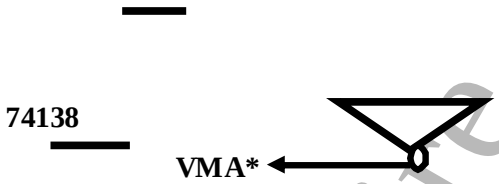


Figure 5.14 Decoder Used for Address Decoding

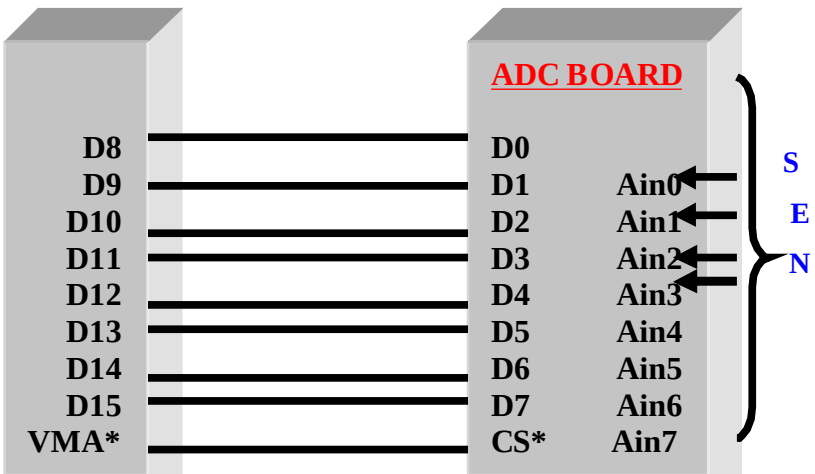
5.7 Experimental setup of the Flexible Development System

The hardware connection of all the blocks discussed above is given in the figure 5.15 which represents the flexible development platform for interactive robots. Single sensor is selected at a time by sending A1, A2, A3 address line data of MC68000 onto selection pins A0, A1, and A2 of ADC board as shown in table. At the same time A21,A22,A23 address line also selected the output Y6 of the 3:8 decoder which is latched and transfer its data at VPA*.

If the microprocessor is ready to take data then it send a signal from VMA* to the CS* which select the ADC chip .Then the physical parameters of the sensor is started to read by the processor and control the speed of the stepper motor via the port B (PB0 pin) of the PI/T MC68230 which is defined as output port.

As no wait state generation circuitry is designed so VPA* and VMA* signals are used rather than DTACK for synchronous data transfer. The zero wait-state time is also less than the access time of the ADC chip. Access time is not a ADC conversion time here.

AS* and UDS* data enables the 3: 8 decoder .Here one sensor works at a time which is chosen by the status of Port A switches of the PI/T.



University of Hertfordshire

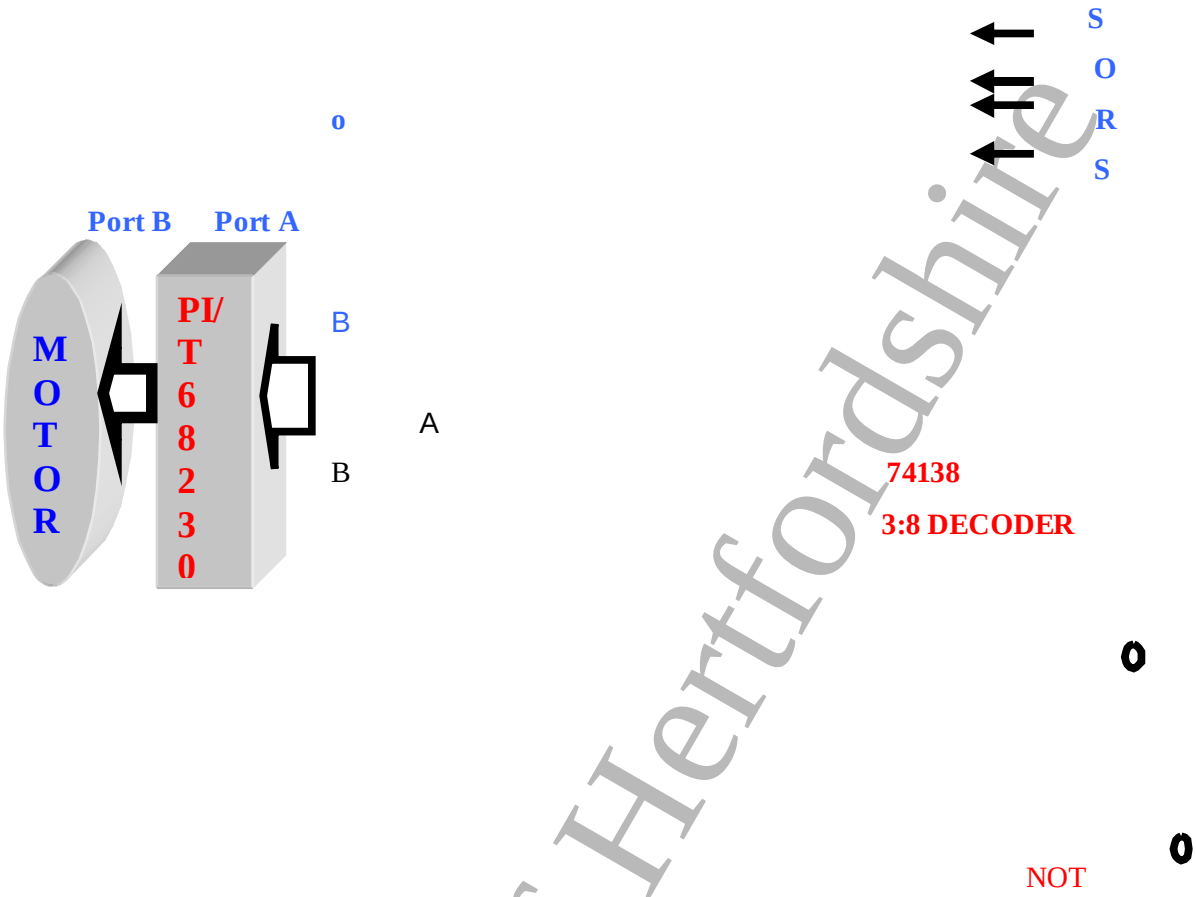


Figure 5.15 : Experimental setup of the whole system

University of Hertfordshire

The status of the switches of Port A of the PI/T which decided that which input of the ADC is worked at present is given as follow:

- If only PA0 switch active and other switches are in-active then only Ain0 sensor works.
- If only PA1 switch active and other switches are in-active then only Ain1 sensor works.

- If only PA2 switch active and other switches are in-active then only Ain2 sensor works.
- If only PA3 switch active and other switches are in-active then only Ain3 sensor works.
- If only PA4 switch active and other switches are in-active then only Ain4 sensor works.
- If only PA5 switch active and other switches are in-active then only Ain5 sensor works.
- If only PA6 switch active and other switches are in-active then only Ain6 sensor works.
- If only PA7 switch active and other switches are in-active then only Ain7 sensor works.

5.8 Summary

This chapter contains all the useful information regarding the hardware part of the flexible development platform. All the parts are discussed in detail which are followed by the experimental setup of this project. The working of all the elements, ADC ,Bus Expander, I/O buffered board, Stepper motor controller ,Address decoder is described step by step and finally the whole setup of this platform is specified for easy understanding of the subsequent of the readers.

CHAPTER 6

SOFTWARE DESIGN

6.1 Introduction

Crossware www.crossware.coom), a leading embedded software tools developer, has released a suite of software for the FLT-68K Microprocessor Trainer kit.

University of Hertfordshire

Crossware Embedded Development Studio includes a re-locatable macro cross-assembler, source level simulation and source level debugging and comprehensive documentation .It contains a sophisticated ANSI C compiler with the features of Windows 95 and Windows NT4.0 to provide a development environment.[10]

The advanced training suite of FLT- 68K used in the project has the following feature[10]

- Optimising ANSI C compiler and libraries (Advanced Edition only).
- Relocatable cross assembler, linker and library manager Source level simulator to debug without hardware.
- Source level serial port debugger to debug with FLT-68K Microprocessor Trainer board via the PC serial port.
- EPROM's containing the Crossware debug monitor to replace the existing FLT-68K monitor.
- Embedded Development Studio - a user friendly integrated development environment which eliminates the learning curve for using the tools.
- Manuals in printed and electronic form.
- Secure license server to simplify multi-user licensing across a LAN.
- Technical support and free up for 2 months. [11]

C languages is chosen because of its many advantages as compared to other Assembler languages and produce efficient code as discussed in chapter 4.Several software design qualities parameters (like flexibility, efficiency, reliability and portability)keep in mind before designing this software program.

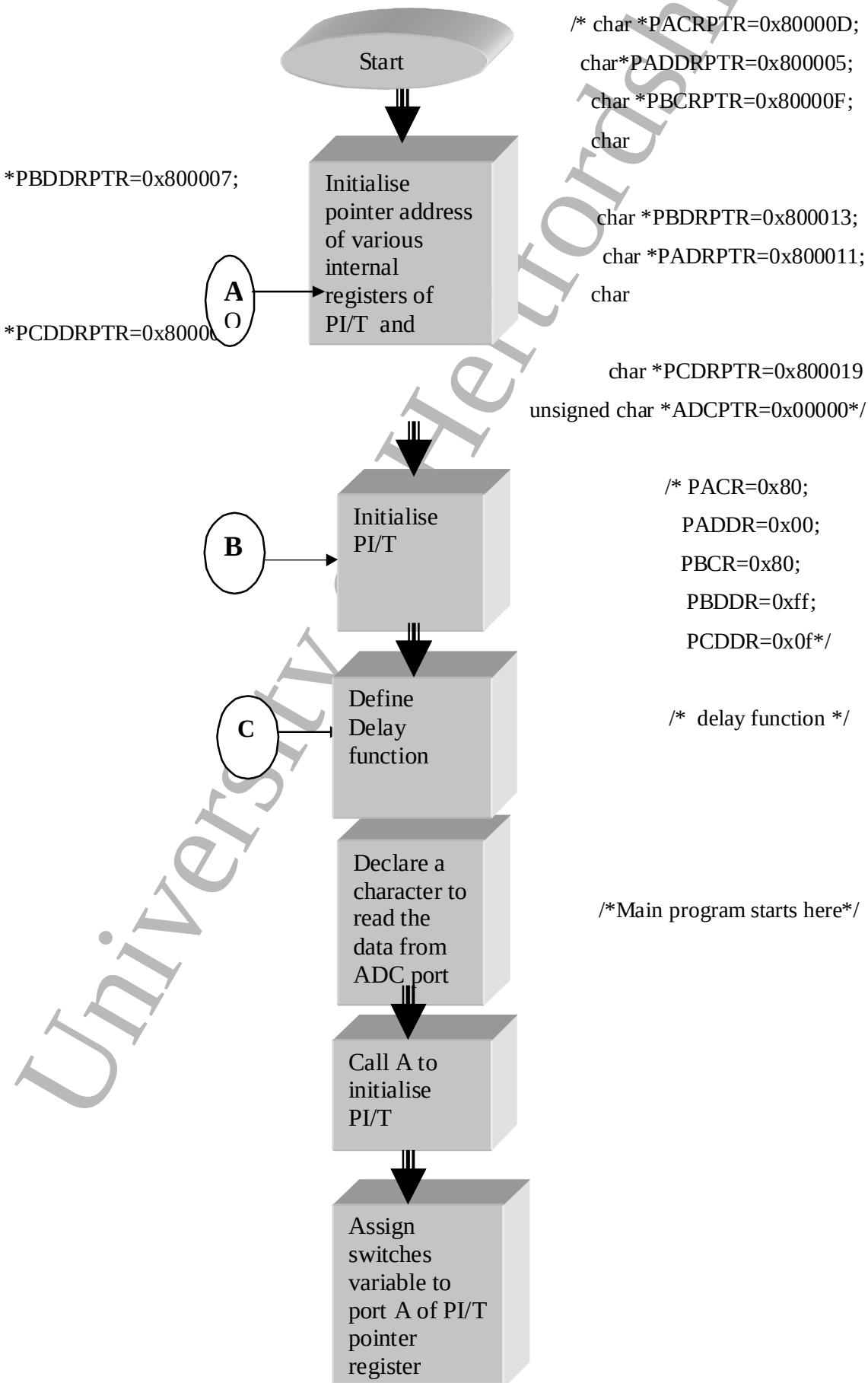
6.2 Software Program

The software program represents in this section includes the overall working of the flexible development platform (refer Appendix for program coding). The program for testing of individual element is given in the chapter 7 (Functional test result).

The step by step working of whole setup by programming is shown in the flow chart (figure 6.1).

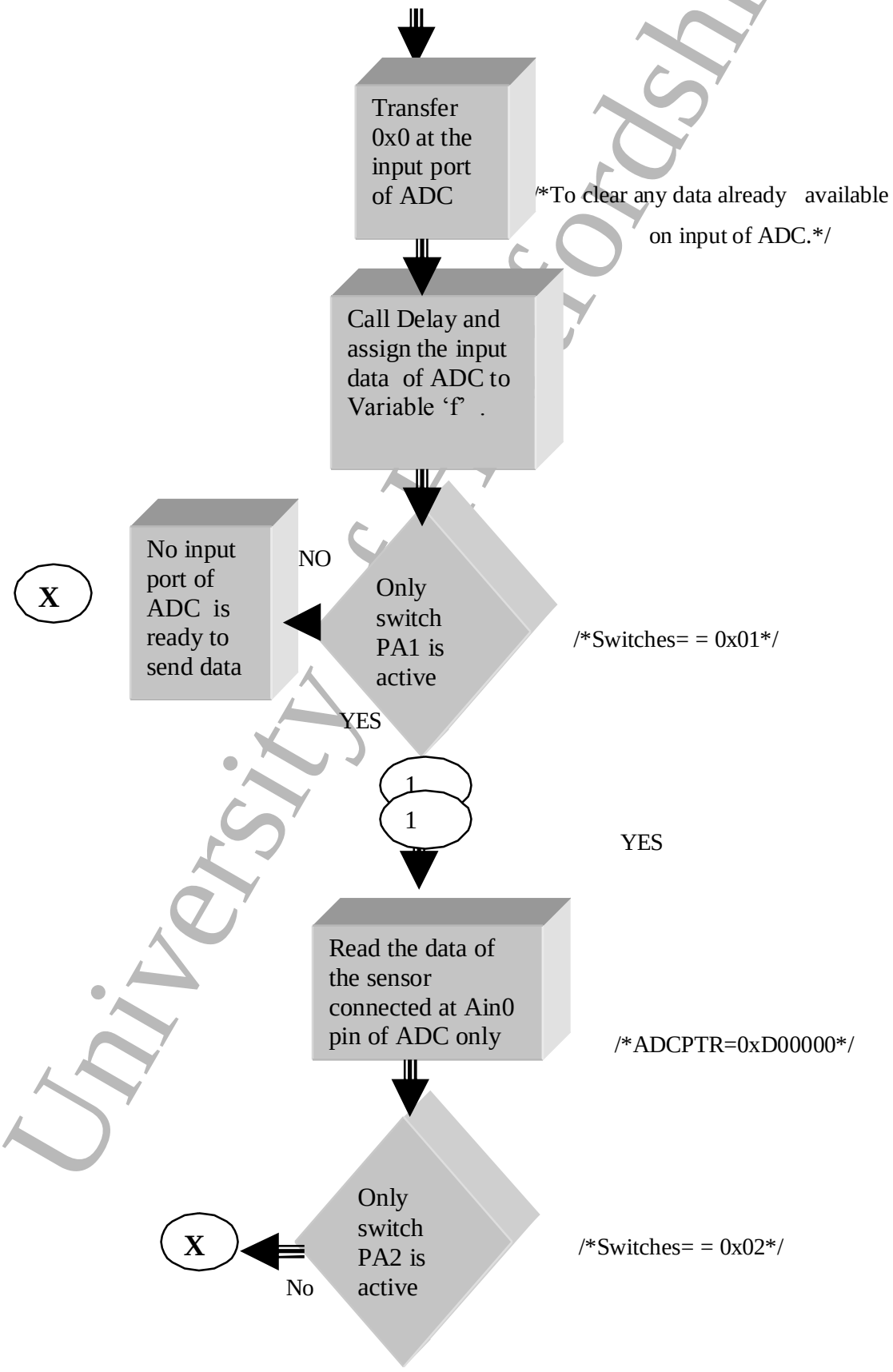
This program includes the way by which individual sensor is chosen from ADC at a time according to the status of the switches of port A of PI/T 68230 and the output is taken from the port B to control the speed of the stepper motor.

University of Hertfordshire

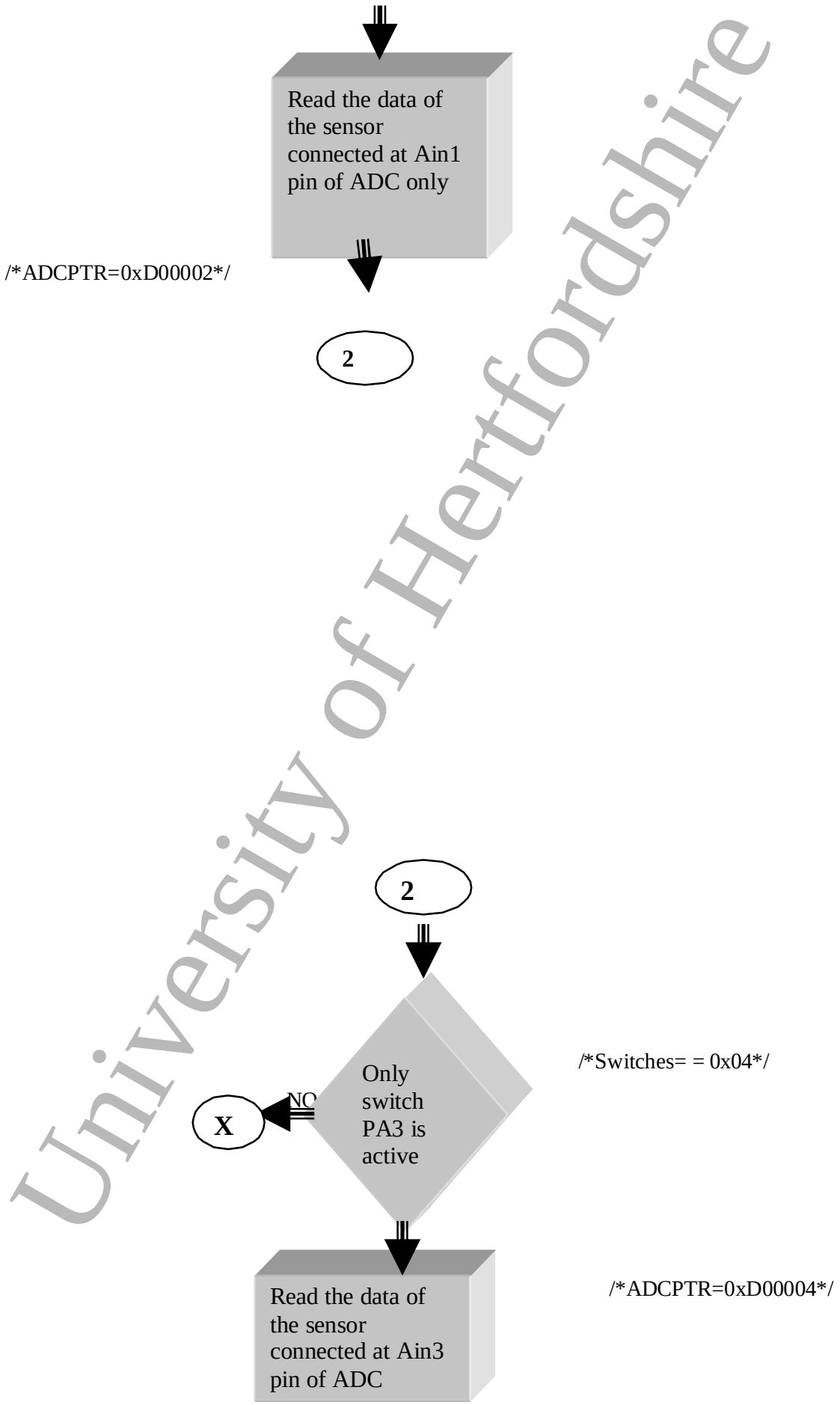


University of Hertfordshire

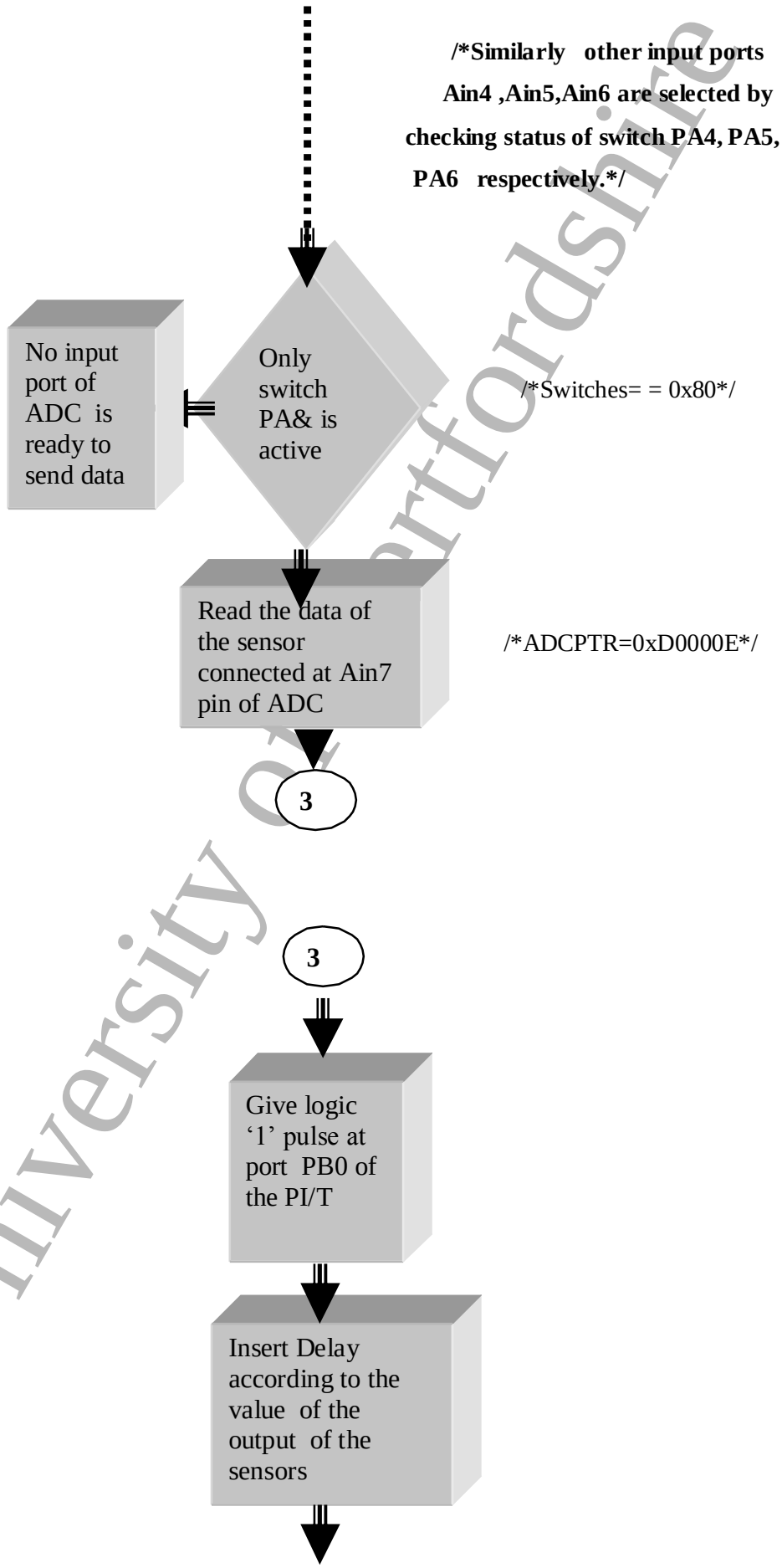
/*Unconditional true loop stats here*/


Transfer 0x0 at the input port of ADC
/*To clear any data already available on input of ADC.*/
Call Delay and assign the input data of ADC to Variable 'f' .
Only switch PA1 is active
/*Switches= = 0x01*/
NO
No input port of ADC is ready to send data
X
YES
1
1
YES
Read the data of the sensor connected at Ain0 pin of ADC only
/*ADCPTR=0xD00000*/
Only switch PA2 is active
/*Switches= = 0x02*/
No
X

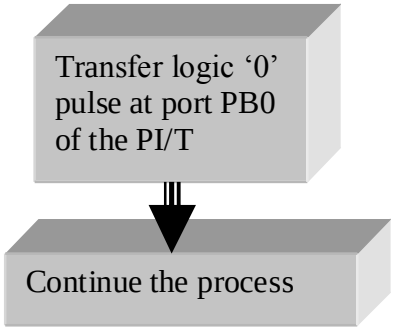
University of Hertfordshire



University of Hertfordshire


/*Similarly other input ports Ain4 ,Ain5,Ain6 are selected by checking status of switch PA4, PA5, PA6 respectively.*/
No input port of ADC is ready to send data
Only switch PA& is active
/*Switches= = 0x80*/
Read the data of the sensor connected at Ain7 pin of ADC
/*ADCPTR=0xD0000E*/
3
3
Give logic '1' pulse at port PB0 of the PI/T
Insert Delay according to the value of the output of the sensors

University of Hertfordshire



/*Logic '1' and '0' pulses are transfer to generate square wave to control the speed of The motor */

Figure 6.1 : Working of the software program

This flow chart make the program very easy to understand and help the reader to elaborate the idea of this project for further development without any problem

6.3 Summary

In this chapter the main software modules is explained which help in making decision that which sensor is chosen at a time to control the speed of the motor or any real world application. The program is simple to understand and the whole program code is present in Appendix

CHAPTER 7

FUNCTIONAL TEST RESULT

7.1 I/O Buffered Board Test

Before using I/O buffered board , its initial testing is necessary by giving logic '0' and '1' pulses at the input port of the PI/T and taken the output from the consecutive pin of the output port.

For example : when PA0 switch is active the LED at the port PB0 glows i.e. the data given at input port A is taken out from the port B

Similarly other pins of the port A and port B tested.

For testing program refer Appendix B.2.

```
char switches;                  // switches is declared to take data from  input po
```

University of Hertfordshire

```
PACR=0x80;                  //  Initialisation of the
PADDR=0x0;
PBCR=0x80;
PCDDR=0x0f;
PBDDR=0xff;
While (1)
{
 Switches=PADR;              // Port A data transfer at variable switches.
PBDR=switches;              //Transfer switches data at port B.
Switches=PCDR;               // upper pins (PC7-PC4) of Port C data
                                 is transfer at variable switches.
Switches=switches>>4;
PCDR=switches;               //Transfer switches data at lower pins (PC0-   PC3) of port C.
}
```

This test is successfully completed and also the working of upper and lower port pins of port C of PI/T is checked

7.2 **Single input ADC Test**

Initially the testing is to be done on the single input Ain0 of the ADC for checking proper functioning of the ADC board. A +5 V signal is given at the Ain 0 pin continuously and this input is transfer at variable 'f' as shown in program below. This analogue voltage is converted into binary '1' and note down from the port B of the PI/T.

If a variable voltage is given at the Ain0 then the output of the port B is changed according to the value of this variable analogue signal. The changes at the port B is easily seen by using the watch window option of the Embedded design software.

```
unsigned char f;
PBCR=0x80;                  // configure the PI/T to work with the Buffered I/O Board
 PBDDR=0xff;
 While (1)
{
 ADC=0x0;                   //Initially transfer 0x0 data at ADC input.
 delay (10);
```

University of Hertfordshire

```
 f =ADC;                    //Assign the ADC input data to variable 'F'
 PBDR = f;                  //Transfer this data on the port B of the PI/T
delay (10);
}
```

This test also done successfully and the program code of the single input ADC test is present in Appendix..B.3

7.3 Multiple Input ADC test

As the main aim of the project is to make the development platform flexible so the different type of the sensors are connected at the eight input pins of the ADC board All these sensors work simultaneously but at a time only one sensors is used to control the speed of the motor. Any sensor chose according to the requirement of the application. For example in the thermal power station , high quality temperature sensor is used to maintain the temperature of the boiler up to the safe level by blowing alarm at the output or automatically switch off the boiler.The testing of the multi inputs working of the ADC is done by the following program :

```
Unsigned char f;
Char switches;        //Declare variable named switches
Include PIT ();      // Initialise PI/T

While (1)
                      {
Switches=PADR;
if (switches==0x01)     //port A data of PI/T(PA0 switch active only)
ADCPTR=0xD00000;    //ADC input A0 works
else
PBDR=0
if (switches==0x02)    //port A data of PI/T(PA1 switch active only)
ADCPTR=0xD00002;     //ADC input A1 works
else
PBDR=1;
if(switches==0x04)     //port A data of PI/T(PA2 switch active only)
ADCPTR=0xD00004;     //ADC input A2 works
if(switches==0x08)     //port A data of PI/T(PA3 switch active only)
```

University of Hertfordshire

```
ADCPTR=0xD00006;     //ADC input A3 works
if (switches==0x10)      //port A data of PI/T(PA4 switch active only)
ADCPTR=0xD00008;     //ADC input A4 works
 if(switches==0x20)      //port A data of PI/T(PA5 switch active only)
ADCPTR=0xD0000A;    //ADC input A5 works
 if (switches==0x40)      //port A data of PI/T(PA6 switch active only)
 ADCPTR=0xD0000C;   //ADC input A6 works
 if (switches==0x80)      //port A data of PI/T(PA7 switch active only)
 ADCPTR=0xD0000E;   //ADC input A7 works
 ADC=0x0;             //initial data of ADC port
 delay (10);
 f=ADC;

PBDR=1;                      //High level signal at port B of PI/T
delay (f);
 PBDR=0;                      //Low level signal at port B of PI/T
 delay (f);

}
```

Testing is finished successfully and all the sensors are worked .Switches at the port A of PI/T easily controlled the one input at a time, the output waveform at the PB0 pin of the port B is shown in figure. The maximum voltage of +4.9V is given by the analogue voltage sensor.

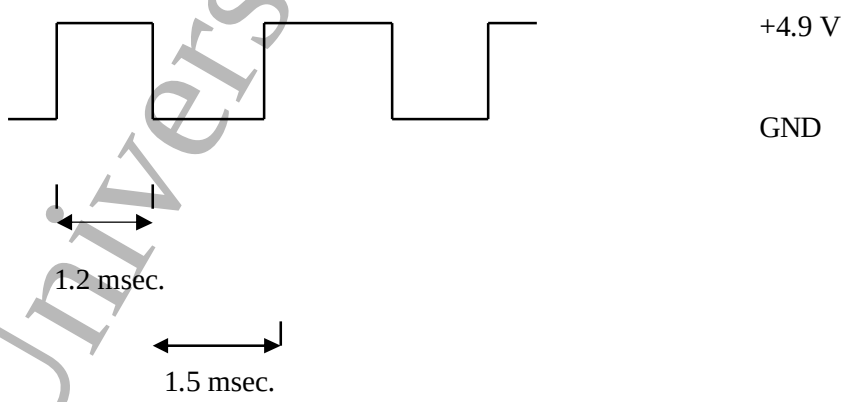


Figure 7.1 Output waveform at Port B by Voltage sensor

The speed of the motor is controlled by the generation of the square waveform

University of Hertfordshire

7.4 Summary

The final testing of individual elements and the whole set up is included in this chapter. Step by step testing of all the modules helped to successfully completed this project.

CHAPTER 8

PROJECT MANAGEMENT

8.1 Overview

For the successful completion of the project, the project time management is a key issue at the beginning of a project. Initially, the project time plan had been elaborate in the feasibility study with regards to the time allocated to the different project stages and submission deadlines defined by the department as described in the *Major Project Information for Students & Staff* handbook. Through out the project this timing schedule is strictly followed.

This chapter includes the initial timing deadlines and the actual time which author spent on this project .As the final time plan is slightly different from the initial time plan therefore certain tasks had been readjusted.

Time Management

Time planning is important in ensuring the progress of the project on track to completion. A good time plan would have a flexibility or allowance of time in completion of every stage of any project that is taken

University of Hertfordshire

Tasks Performed	Initial Time Planned	Actual Time Spent
*Choose Sensors and Motors	8 Weeks	10 Weeks
*Hardware Circuit Design	4 Weeks	8 Weeks
*Software Learning	4 Weeks	4 Weeks
*Interfacing with 68000	4 Weeks	5 Weeks
*Troubleshooting and Testing	8 Weeks	8 Weeks

Table 8.1 : Time table of the important steps of the project

At the start of the project ,initial time plan Gantt chart was drawn up in the feasibility report but that time plan is little bit modified (not in term of time scale but only drawing) for easy comparison with the final time plan Gantt chart as shown in Figure and Figure . A comparison of the original time plan and the actual time plan clearly shows that the testing of the different sensors took a little longer than was originally estimated. This is due to the more time devoted to do experiment on DCP sound sensor and three sensors of Lego Mindstorms for using in this project. Four weeks were spent on doing research on finding function of different pins of sound sensor and time also spent on contacting DCP commotion group (manufacturer).But no important information was found which helped in this project. So author decided to design his own sensors to develop a flexible platform.

University rdshire

The technical help for design were answered swiftly but were not always helpful with regards to the questions asked about this project. This explains the extension in time while researching stopping.

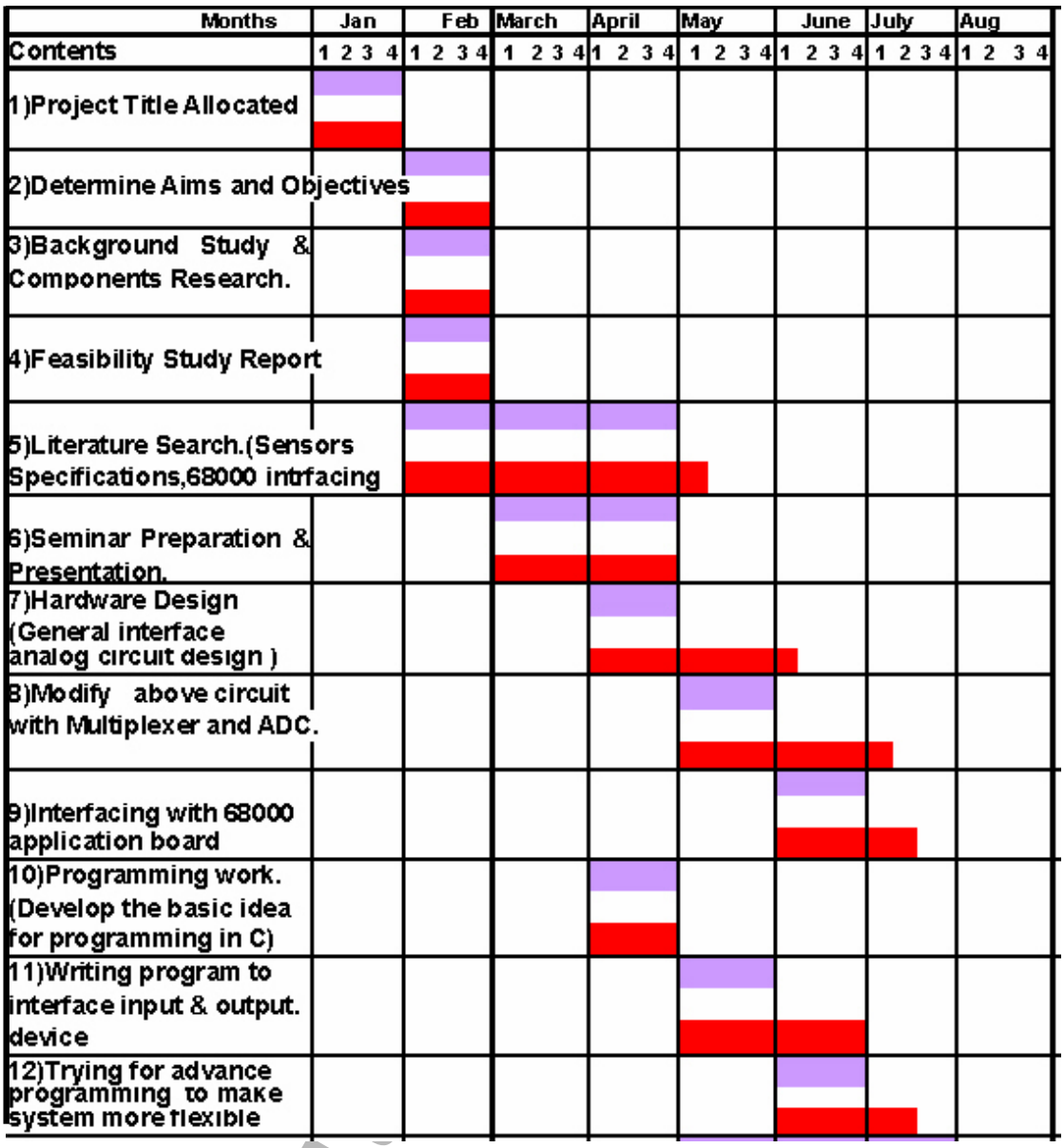
The period of time in which the tests for sensor were completed was delayed as a result of the extension in the research to find the sensor in which output and input were taken from different wires unlike sound ,touch and distance sensor of Lego Mindstorms.

Due to this, the development of the sensor started later than anticipated. This task ran into frequent difficulties, as mentioned earlier in chapter 10. Due to these factors, the sensor electronics did not reach its full potential, but this will allow further work to be carried out.

The design for this project was kept as simple as possible and as a consequence of this, the construction time was less than had been estimated. The period of time left for debugging and troubleshooting was also extended, but in general it can be seen that this project progressed steadily and kept to plan.

Figure 8.1 shows the initial Gantt (Blue Color) chart created at the start of the project. Efforts were made to adhere to the initial Gantt chart throughout the project but where this became impossible the Gantt chart was updated. The final Gantt chart (Red Color)at the end of the project, following all updates is shown in figure which shows the actual time spent on this project.

University of Hertfordshire


Months
Jan
Feb
March
April
May
June
July
Aug
Contents
1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
1)Project Title Allocated
2)Determine Aims and Objectives
3)Background Study & Components Research.
4)Feasibility Study Report
5)Literature Search.(Sensors Specifications,68000 intrfacing
6)Seminar Preparation & Presentation.
7)Hardware Design (General interface analog circuit design)
8)Modify above circuit with Multiplexer and ADC.
9)Interfacing with 68000 application board
10)Programming work. (Develop the basic idea for programming in C)
11)Writing program to interface input & output. device
12)Trying for advance programming to make system more flexible

Universi

Gantt chart continues on next page…..

…..continue

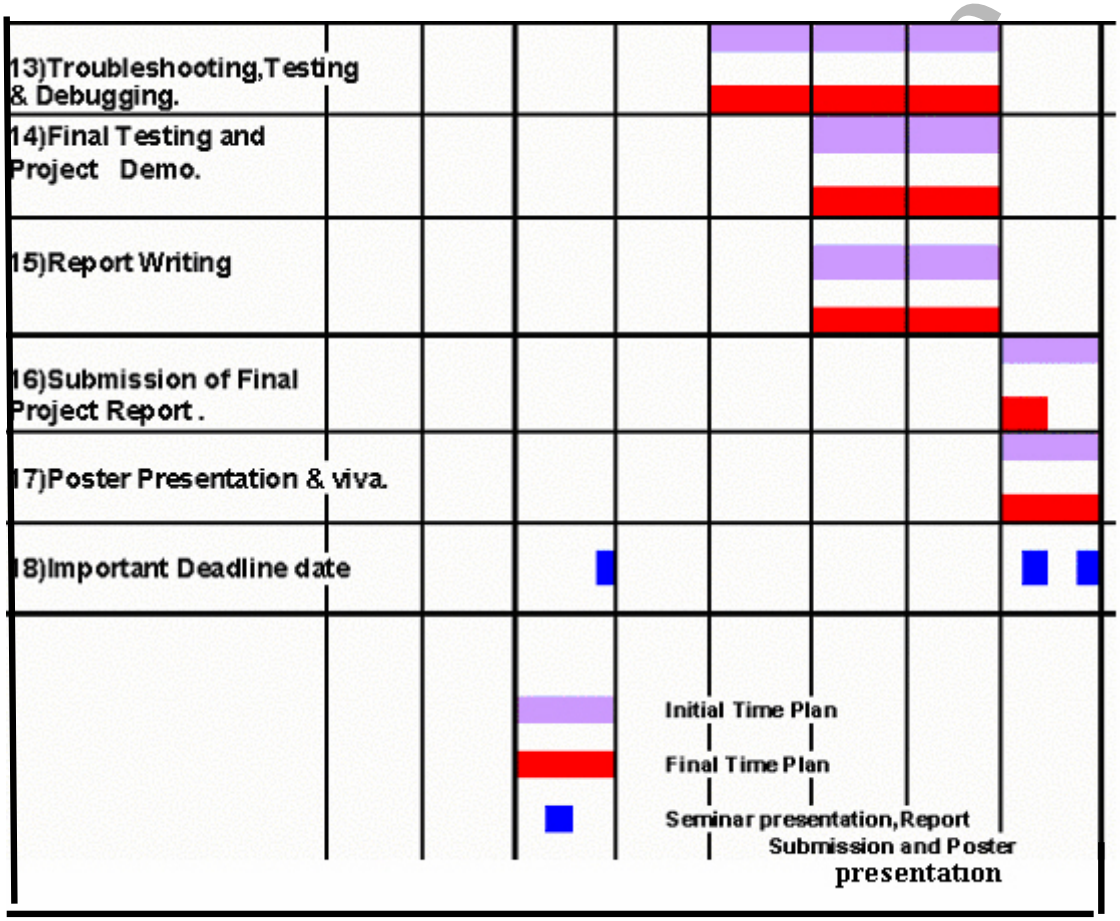


Figure 8.1 Initial and Final time plan Gantt chart

8.3 Summary

This chapter includes the initial time set up by the author and the actual time spent on this project with reasons for time difference between the both Gantt chart.

University

CHAPTER 9

CONCLUSION AND FURTHER DEVELOPMENT

9.1 Introduction

This chapter split into split into two sections, the first is about what the project achieved and the conclusion surrounding this. The second is about further work which has been carried out if the time allowed or if the work will be continue by somebody else.

9.2 Conclusion

The project describes the design and performance of the micro processor based robot controller The knowledge of the programming language C and hardware was very useful to carry out the project, most of the problems encountered in the project is how to design the sensors used at the input port of the ADC.

The initial objective of using sensors of Lego Mindstorms and Logit is not fulfilled because after deep study in sensors it is concluded that all these sensors are designed according to their manufacturer companies software So all the sensors used by author itself by which he gains a valuable knowledge in the fascinating field of sensors , actuators and Microprocessors. The sensors circuit was tested and proved to function correctly, according to their design. In doing this project, the author learned how to use a microprocessor 68000 from both hardware and software approach. When dealing with this project, it is found that it could not always simply follow the theoretical ideas due to physical limitations, everything seems to be working fine could also go wrong at any time, so it must prepare more time for testing and debugging.

The flexible development platform is designed successfully but it has some drawbacks as well. In this platform ,the sensors output is changed by the physical parameters of the environment and this analogue data is given at the input terminal of the ADC to convert it into digital value.

Than this data is synchronously transfer at the output port B of the PI/T 68230 by using VMA*(Valid Memory Address) and VPA*(Valid Peripheral Address) signals of MC68000 to control the speed of the stepper motor connected at the port B (PB0).

University of Hertfordshire

The use of C language programming also increases the reliability and functionality of the program. This platform has many real world applications like Temperature Control ,Air Quality Monitoring, Traffic Counting ,Energy Management etc..

However even after the understanding of the subjects of Microprocessor and sensors, this project has one drawback that only one sensors is active at a time which is controlled only by the status of the switches of the input port A of the PI/T68230.But this drawback is easily eliminated as discussed in the further development.

Much of the time is devoted in the study of the different types of the sensors which disturbed the time plan of the project. By giving lots of effort in this project ,this project is completed on time.

9.3 Further Development

A lot of necessary groundwork is to be done. Therefore if anybody want to continue this project, they would already have a good understanding of what required and what problem have to be solved by studying this project report. There are many ways by which further modification of this project is possible, such as:

- **Sensors:** The high quality may be used at the input of the ADC which give the quick response as the environmental condition changes makes the system more flexible because sensors are the only devices that supply external, environmental information to the robotic system.

- **Address Decoding:** For interfacing with the Microprocessor address decoding is to be done with the advance technology of FPGA (Field Programmable Logic Array),PLA (Programmable Logic Array) and PAL(Programmable Array Logic) instead of the address decoder. These devices share the random logic elements and ROM based decoder properties which gives optimum design from the point of view of the speed and provides flexibility and compactness at the cost of a slightly slower speed and a restriction on the number of variables that can be handled economically.

- **Multiple output :** More than one motor can be easily used by using the timers of the PI/T(Program Interface/Timer) 68230 or by designing the motor controlled circuit which able to control the working of more than one motor simultaneously.

University of Hertfordshire

University of Hertfordshire

REFERENCES:

[1] http://www.robotbooks.com/Lego-Mindstorms.htm

[2]. LEGO® MINDSTORM™**:** ***Robotics Invention System 1.5***, © 1999 the LEGO Group

[3] http://research.cs.tamu.edu/prism/lectures/iss/iss_l2.pdf

[4] http://www.plazaearth.com/usr/gasperi/lego.htm#basics

[5] www.dcpmicro.com

[6] MC68000 8-/16-/32-Bit Microprocessors User's Manual, MC68000UM, 9th Ed Motorola Inc, p11-22

[7] http://www.flite.co.uk/micros/flt-68k.html

[8] http://balder.prohosting.com/sywu/Clang_1.htm

[9] http://www.seattlerobotics.org/encoder/jul97/basics.html (basicof sensor

[10] http://www.crossware.com/press/october01/flt68kts.html]

[11]http://www.crossware.com/datasheets/flt68kts.pdf

[12] Clements Alan, Microprocessor System Design: *68000 hardware, software and interfacing*, 3rd edition, Boston, Mass; London: PWS Publication,1997, .

[13] Mini –Projects (2ELE0047) Reference manual in LD403 lab provided by Dr. David Lee

[14]http://www.acfr.usyd.edu.au/teaching/4th-year/mech4721-Signals/material/lectures/sensors/1-Introduction.pdf

[15] http://www.legomindstorms.com/internals/

BIBLIOGRAPHY

1. Walter, Triebel A and Singh, Avtar: The *68000 and 68020 Microprocessors*, Prentice-Hall, 1991

2. http://dragon.herts.ac.uk/labs/c460/Components

3 http://www.legomindstorms.com/

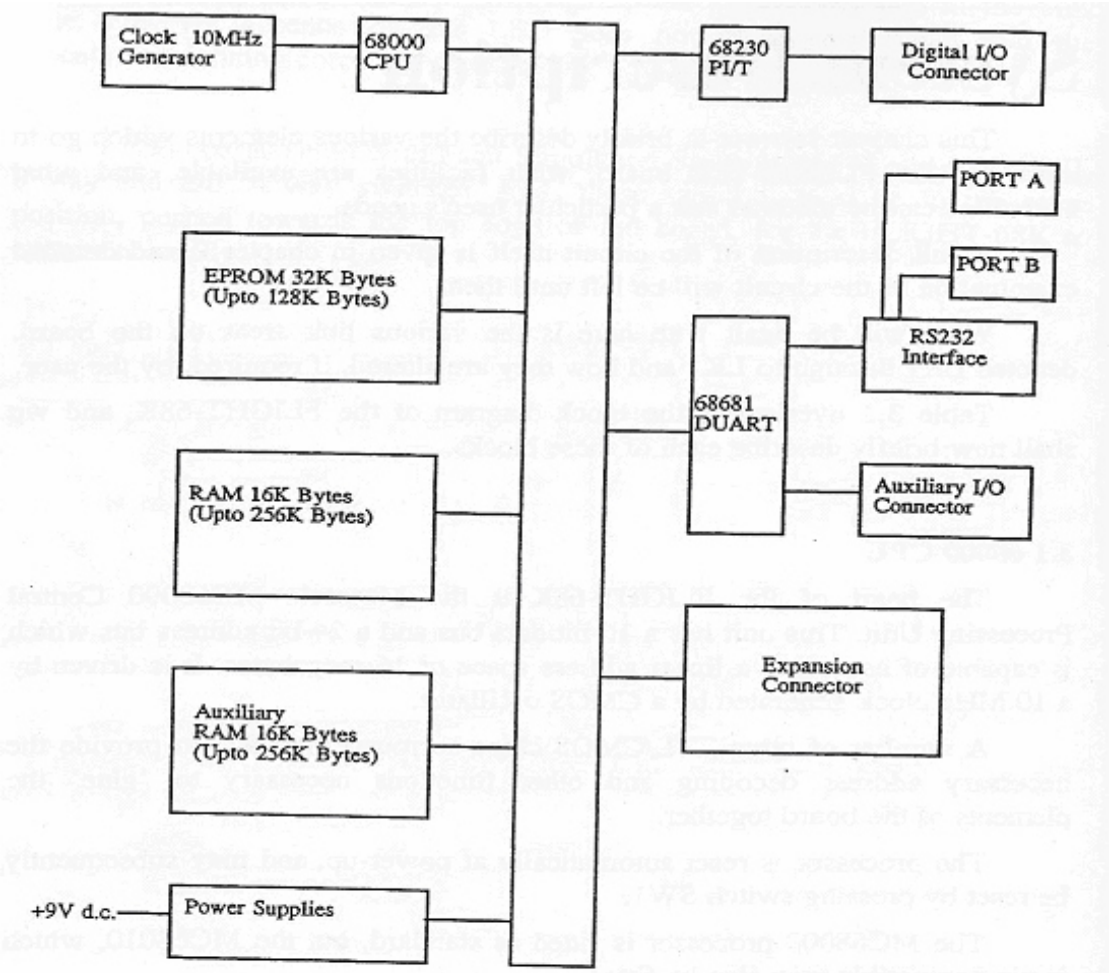
4 .Kleitz, William, *Digital and Microprocessor Fundamentals*,2nd Edition 1997

5.H.R Everett, "Sensors for Mobile Robots", *Theory and Application*, A K Peters, Ltd,1995

University of Hertfordshire

University of Hertfordshire

APPENDIX A.1: BLOCK DIAGRAM OF THE FLIGHT -68K BOARD



Block Diagram of the FLIGHT 68K

Universi

APPENDIX A.2: SPECIFICATIONS OF THE FLIGHT -68K BOARD

APPENDIX L FLIGHT-68K SPECIFICATION

PROCESSOR: Motorola MC68000P10

CLOCK SPEED: 10MHz

EPROM: up to 128k bytes

RAM: up to 256k bytes

AUXILIARY MEMORY: RAM - up to 256k bytes

PERIPHERAL EXPANSION: Processor Bus

POWER: +9 volts, unregulated, at 700mA (Consumption may vary if memory configuration altered or expansion card added)

SYSTEM FIRMWARE: 64k bytes of EPROM, comprising comprehensive monitor, line assembler, disassembler, 28 system calls.

INPUT/OUTPUT CAPABILITIES:

Serial I/O: Two RS232 channels, software selectable baud rates of 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19200 baud.Default format: 8 data bits, no parity, one stop bit.

DUART Auxiliary I/O lines:

Input: 4
Characteristics:

	Min.	Max.
Input High Voltage	2.0V	
Input Low Voltage		0.8V
Input Leakage Current (Vin = 0 to Vcc)	-10uA	10uA

Output: 6
Characteristics:

	Min.	Max.
Output High Voltage, excluding Open-collector o/p's (Ioh = -400uA)	2.4V	
Output Low Voltage (Iol = 2.4mA)		0.4V

APPENDIX A.2 (continue) SPECIFICATIONS OF THE FLIGHT -68K BOARD

Univ

Peripheral Interface/Timer (PI/T) I/O Lines:

Port A (PA0 - PA7):	8 I/O lines
Port B (PB0 - PB7):	8 I/O lines
Port C (PC0 - PC7):	8 I/O lines
H1, H3 :	2 handshake inputs
H2, H4 :	2 handshake I/O lines

Characteristics:

	Min.	Max.
Input High Voltage	Vss+2.0V	Vcc
Input Low Voltage	Vss-0.3V	Vss+0.8V
Input Leakage Current (Vin = 0-5.25V)		
H1, H3		10uA
Hi-Z Input Current (Vin = 0.4-2.4v)		
H2, H4, PA0-7, PB0-7	-0.1mA	-1.0mA
PC0-7		20uA
Output High Voltage		
(Iload = -150uA, Vcc = min)		
H2, H4, PA0-7, PB0-7	Vss+2.4V	
(Iload = -100uA, Vcc = min)		
PC0-7	Vss+2.4V	
Output Low Voltage		
(Iload = 2.4mA, Vcc = min)		
All Outputs		0.5V

University C

APPENDIX B.1: Program Code for Whole Flexible Development Platform

/* CREATED BY GULWINDER SINGH*/

void include_PIT();

define ADC (*ADCPTR) //ADC pointer defined

```
define ADC1 (*ADC1PTR)
unsigned char *ADCPTR;
/unsigned char *ADC1PTR;
char *PACRPTR=0x80000D;
char *PADDRPTR=0x800005;
char *PBCRPTR=0x80000F;
char *PBDDRPTR=0x800007;
char *PBDRPTR=0x800013;
char *PADRPTR=0x800011;
char *PCDDRPTR=0x800009;
char *PCDRPTR=0x800019;
#define PACR (*PACRPTR)
#define PADDR (*PADDRPTR)
#define PADR (*PADRPTR)
#define PBDDR (*PBDDRPTR)
 #define PBDR (*PBDRPTR)
#define PBCR (*PBCRPTR)
#define PCDDR (*PCDDRPTR)
#define PCDR (*PCDRPTR)

/*Program Code for Whole Flexible Development Platform*/

void delay(int del)     //define delay function
{
      int f,g;
for(f=0;f<del;f++);
}
main()
{
 unsigned char f;
char switches;
include_PIT();
 /*Senor is selected according to status of the switch of the port A of PI/T
while(1)
```

University of Hertfordshire

```
{
 switches=PADR;
 if(switches==0x01)     //port A data of PI/T
ADCPTR=0xD00000;            //ADC input A0 works
else
PBDR=0;
 if(switches==0x02)    //port A data of PI/T
 ADCPTR=0xD00002;     //ADC input A1 works
Else
PBDR=1;
 if(switches==0x04)     //port A data of PI/T
ADCPTR=0xD00004;            //ADC input A2 works
if(switches==0x08)     //port A data of PI/T
 ADCPTR=0xD00006;            //ADC input A3 works
 if(switches==0x10)     //port A data of PI/T
ADCPTR=0xD00008;            //ADC input A4 works

/*Program Code for Whole Flexible Development Platform*/
 if(switches==0x20)     //port A data of PI/T
 ADCPTR=0xD0000A;          //ADC input A5 works
 if(switches==0x40)     //port A data of PI/T
ADCPTR=0xD0000C;           //ADC input A6 works
 if(switches==0x80)          //port A data of PI/T
 ADCPTR=0xD0000E;           //ADC input A7 works
 ADC=0x0;                    //initial data of ADC port
 delay(10);
 f=ADC;
 PBDR=1;                     //High level signal at port B of PI/T
 delay(f*1
PBDR=0;                     //low level signal at port B of PI/T
 delay(f*1);
}
}
void include_PIT()
```

University of Hertfordshire

```
{
        PACR=0x80;
        PADDR=0x00; PBCR=0x80;
 PBDDR=0xff;
 PCDDR=0x0f;
}
```

APPENDIX B.2 . Program code for testing I/O buffered board.

```
/ 680X0 Initial C Source File
//  Created by   Gulwinder Singh//
//#include "PIT.h"
char *PACRPTR=0x80000D;
char *PADDRPTR=0x800005;
char *PBCRPTR=0x80000F;
char *PBDDRPTR=0x800007;
char *PBDRPTR=0x800013;
char *PADRPTR=0x800011;
char *PCDDRPTR=0x800009;
```

University of Hertfordshire

```
 char *PCDRPTR=0x800019;
 #define PACR (*PACRPTR)
#define PADDR (*PADDRPTR)
#define PADR (*PADRPTR)
#define PBDDR (*PBDDRPTR)
 #define PBDR (*PBDRPTR)
#define PBCR (*PBCRPTR)
#define PCDDR (*PCDDRPTR)
#define PCDR (*PCDRPTR)
char *PCDDRPTR=0x800009;
char *PCDRPTR=0x800019;
#define PCDDR (*PCDDRPTR)
#define PCDR (*PCDRPTR)
 main()
{
char switches;



/*Program code for testing I/O buffered board*/

PACR=0x80;
PADDR=0x0;
PBCR=0x80;
PCDDR=0x0f;
PBDDR=0xff;
while(1)
     {
Switches=PADR;              // Port A data transfer at variable switches.
PBDR=switches;             //Transfer switches data at port B.
Switches=PCDR;              // upper pins (PC7-PC4) of Port C data
                                 is transfer at variable switches.
Switches=switches>>4;
PCDR=switches;             //Transfer switches data at lower pins (PC0-   PC3) of port C.
}
```

University of Hertfordshire

APPENDIX B.3 Program Code for Testing Single Input of the ADC Board

```
/* GULWINDER SINGH*/
#include "PIT.h"
void include_PIT();
define ADC (*ADCPTR)
unsigned char *ADCPTR=0xD0000;
char *PACRPTR=0x80000D;
char *PADDRPTR=0x800005;
char *PBCRPTR=0x80000F;
char *PBDDRPTR=0x800007;
char *PBDRPTR=0x800013;
char *PADRPTR=0x800011;
char *PCDDRPTR=0x800009;
char *PCDRPTR=0x800019;

#define PACR (*PACRPTR)
#define PADDR (*PADDRPTR)
#define PADR (*PADRPTR)
#define PBDDR (*PBDDRPTR)
```

University of Hertfordshire

```
#define PBDR (*PBDRPTR)
#define PBCR (*PBCRPTR)
//#define PCDDR (*PCDDRPTR)
//#define PCDR (*PCDRPTR)
char *PCDDRPTR=0x800009;
char *PCDRPTR=0x800019;

 #define PCDDR (*PCDDRPTR)
 #define PCDR (*PCDRPTR)

 void delay(int del)
 {
       int f,g;

     for(f=0;f<del;f++);
 }

 main()
 {

  unsigned  char f;
```

/*Program Code for Testing Single Input of the ADC Board*/

```
PBCR=0x80;                  // configure the PI/T to work with the Buffered I/O Board
PBDDR=0xff;
While (1)
{
ADC=0x0;                    //Initially transfer 0x0 data at ADC input.
delay (10);
f =ADC;                     //Assign the ADC input data to variable 'F'
PBDR = f;                   //Transfer this data on the port B of the PI/T
delay (10);
}
```

University of Hertfordshire

APPENDIX C : Registers of PI/T 680230

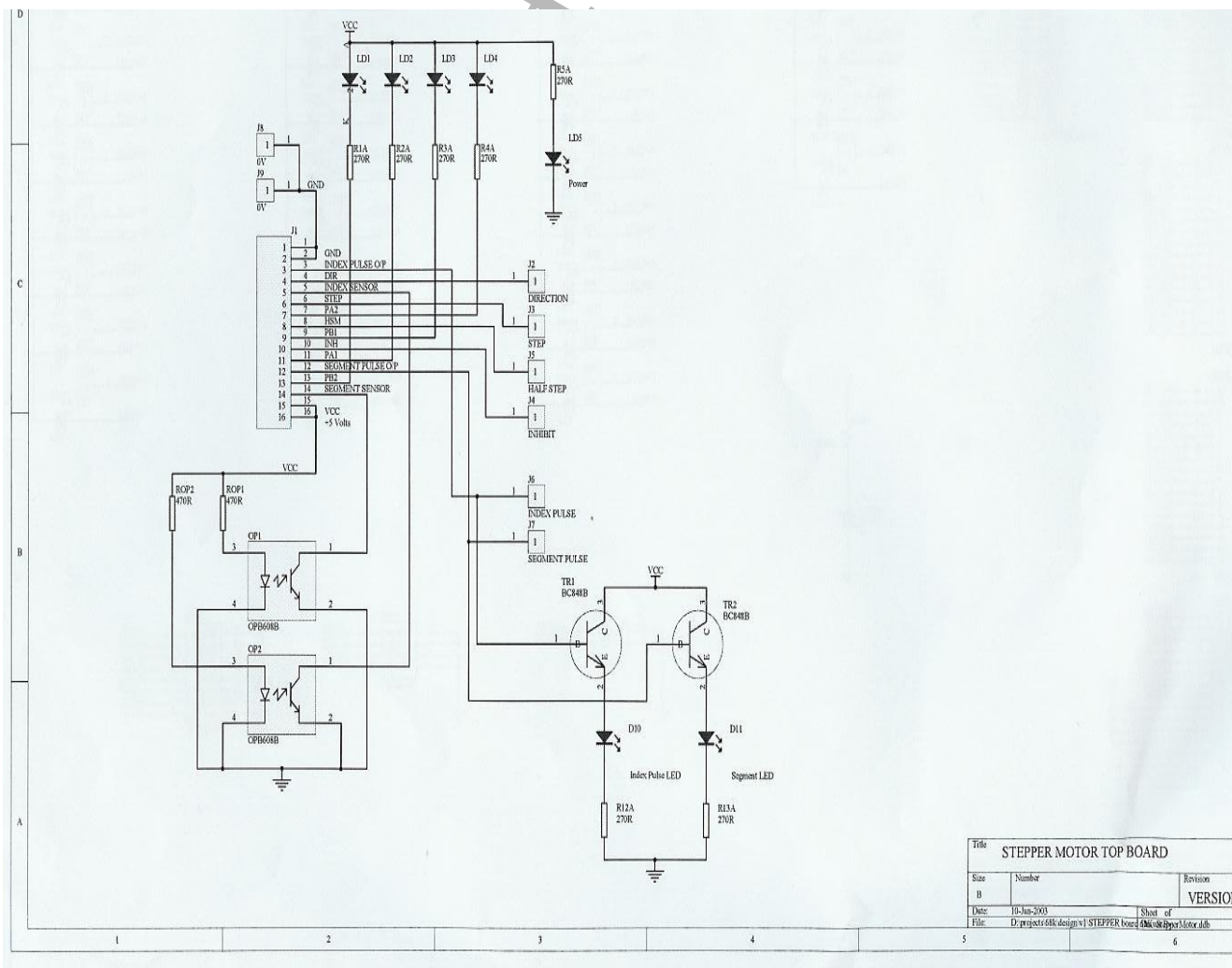
Register select bits							
RS5	RS4	RS3	RS2	RS1	RESISTOR Mnemonic	Register Description	Type
0	0	0	0	0	PGCR	Port general control register	R/w
0	0	0	0	1	PSRR	Port service request register	R/w
0	0	0	1	0	PADDR	Port A data direction register	R/w
0	0	0	1	1	PBDDR	Port B data direction register	R/w
0	0	1	0	0	PCDDR	Port C data direction register	R/w
0	0	1	0	1	PIVR	Port interrupt vector register	R/w
0	0	1	1	0	PACR	Port A control register	R/w
0	0	1	1	1	PBCR	Port B control register	R/w
0	1	0	0	0	PADR	Port A data register	R/w
0	1	0	0	1	PBDR	Port B data register	R/w
0	1	0	1	0	PAAR	Port A alternate register	R/only
0	1	0	1	1	PBAR	Port B alternate register	R/only
0	1	1	0	0	PCDR	Port C data register	R/w
0	1	1	0	1	PSR	Port status register	R/w
1	0	0	0	0	TCR	Timer control register	R/w
1	0	0	0	1	TIVR	Timer interrupt vector register	R/w

University of Hertfordshire

1	0	0	1	1	CPRH	Counter pre load register high	R/w
1	0	1	0	0	CPRM	Counter pre load register med.	R/w
1	0	1	0	1	CPRL	Counter pre load register low	R/w
1	0	1	1	1	CNTRH	Counter register high	R/only
1	1	0	0	0	CNTRM	Counter register medium	R/only
1	1	0	0	1	CNTRL	Counter register low	R/only
1	1	0	1	0	TSR	Timer status register	R/w

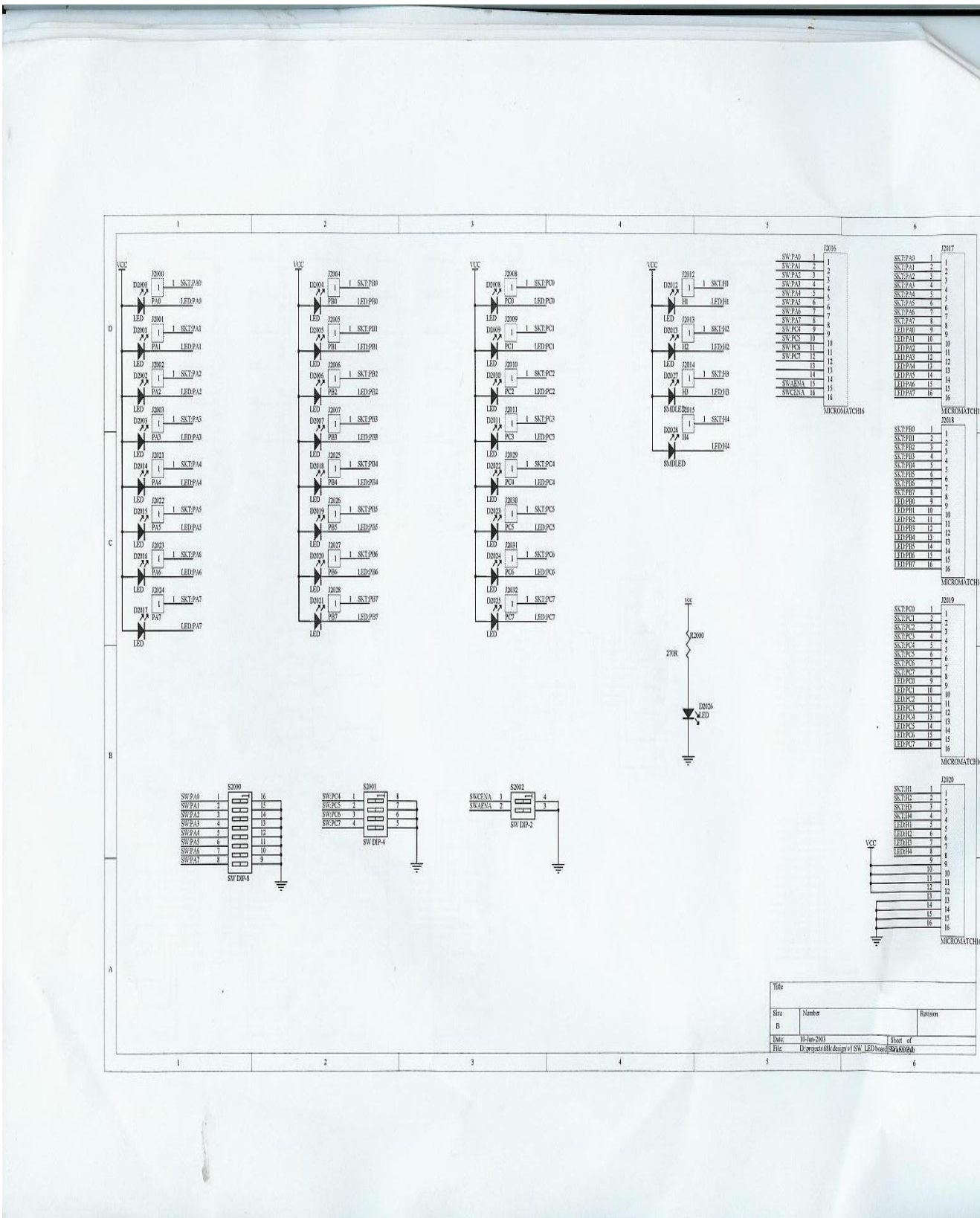
Hertfordshire

APPENDIX D : Stepper Motor Board Circuit Diagram

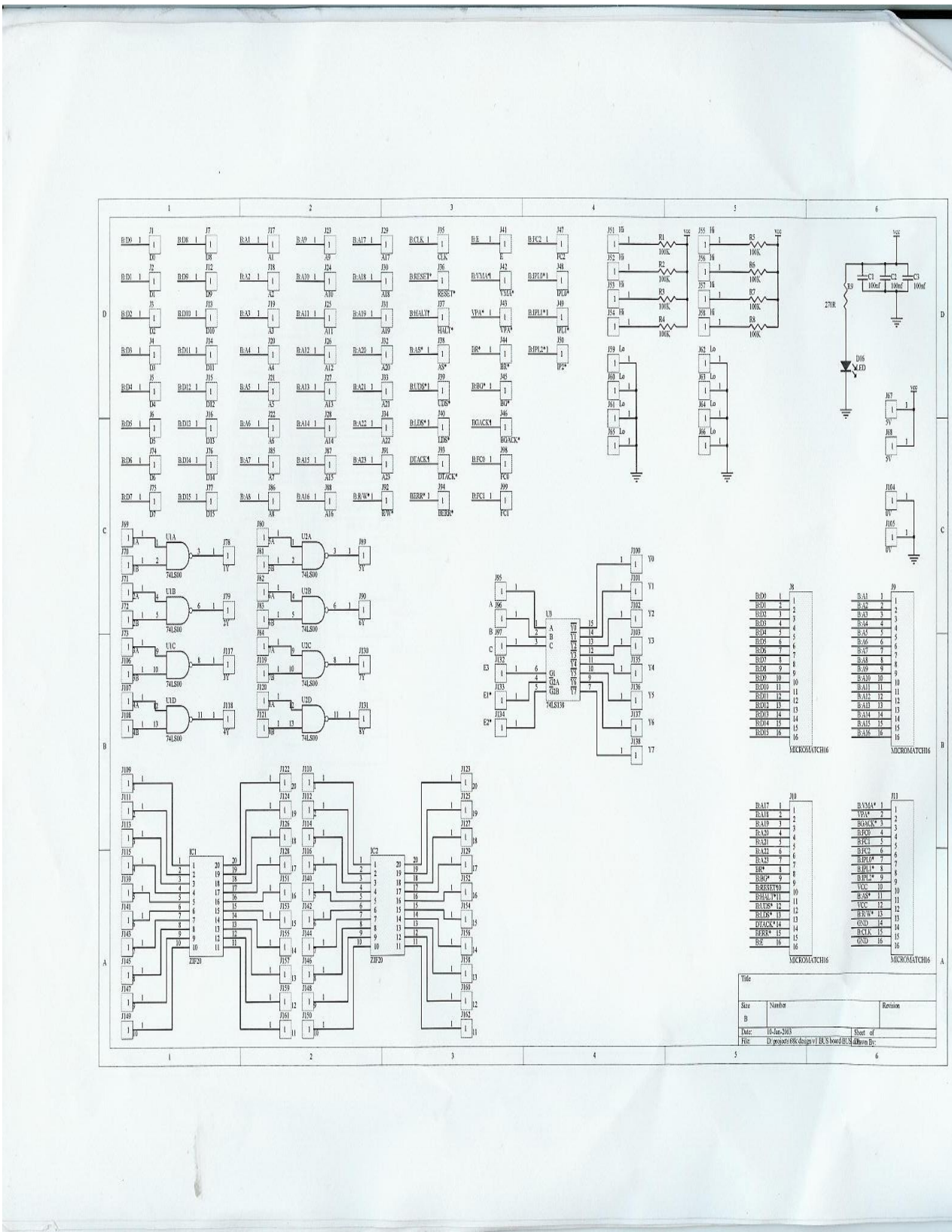


APPENDIX E : I/O Buffered Board Internal Circuit Diagram

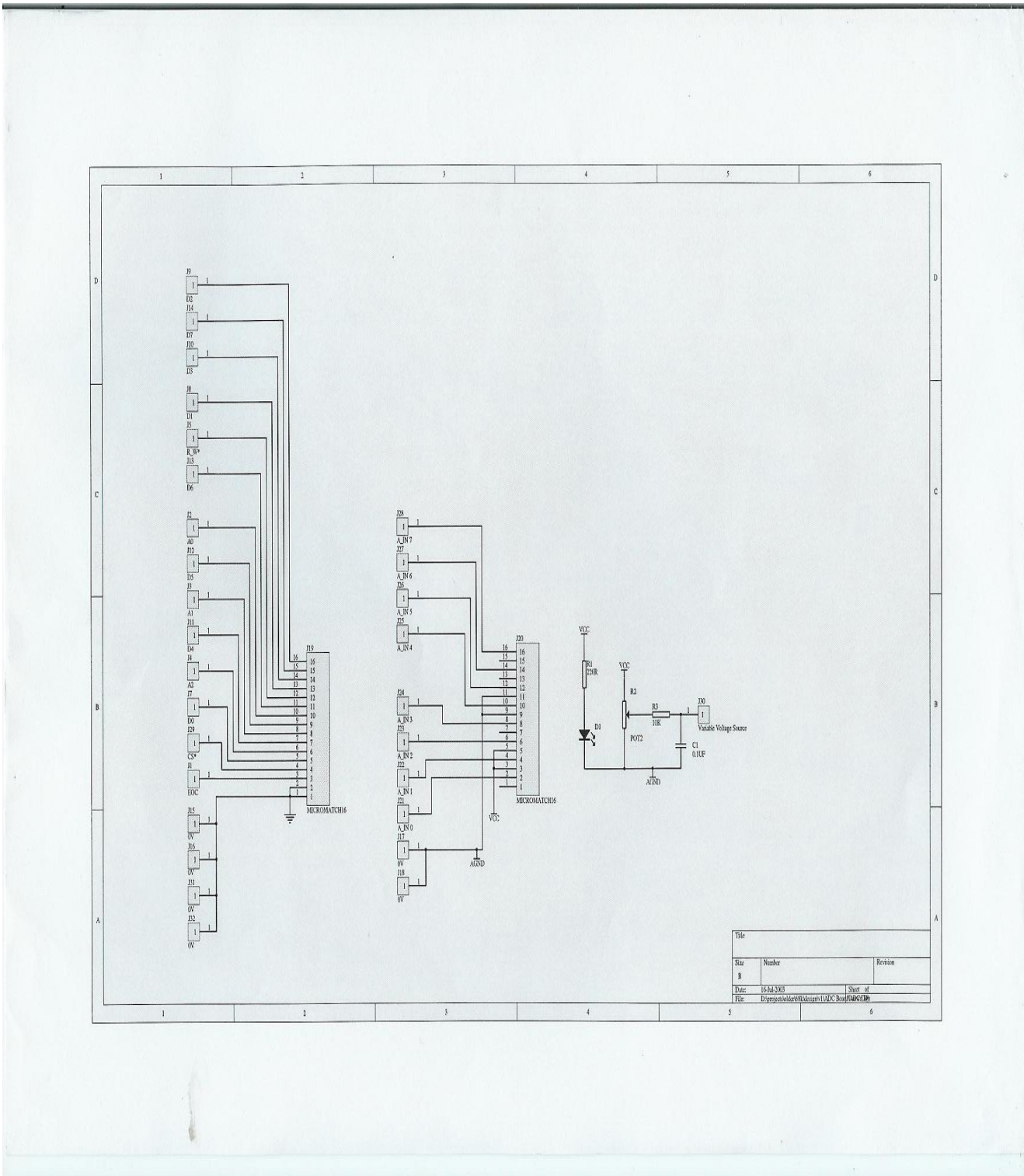
University of Hertfordshire



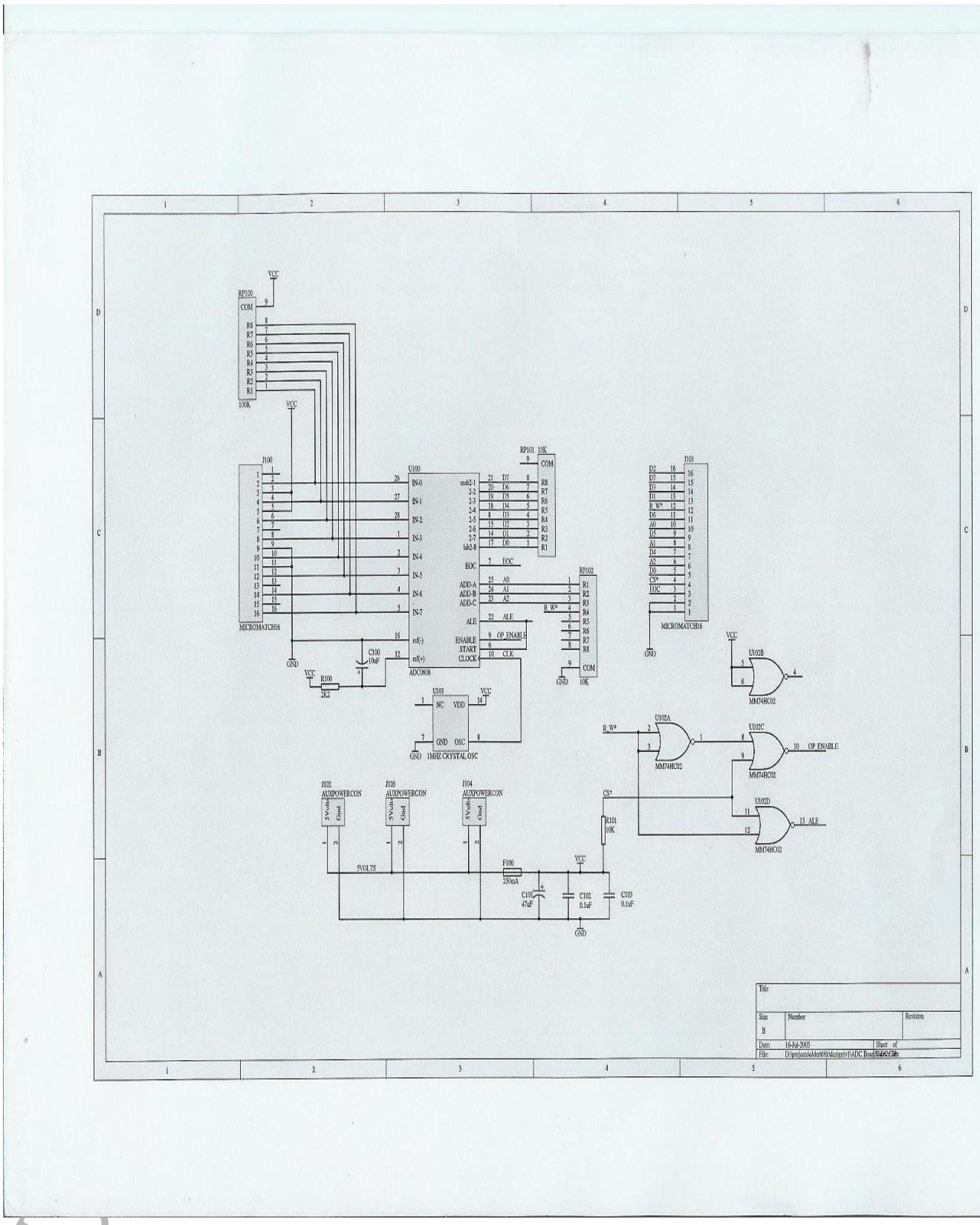
APPENDIX F : Bus Expander Board Internal Circuit Diagram



APPENDIX G.1 : ADC Board Internal Circuit Diagram



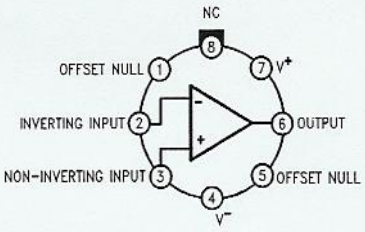
APPENDIX G.2 : ADC Board Internal Circuit Diagram



APPENDIX H.1: Connection Diagram Of Op-Amp 741

Connection Diagrams

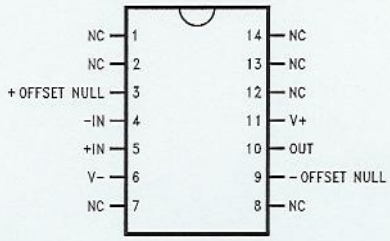
Metal Can Package



TL/H/9341–2

Order Number LM741H, LM741H/883*, LM741AH/883 or LM741CH
See NS Package Number H08C

Ceramic Dual-In-Line Package

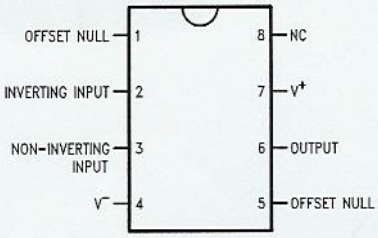


TL/H/9341–5

Order Number LM741J-14/883*, LM741AJ-14/883**
See NS Package Number J14A

*also available per JM38510/10101
**also available per JM38510/10102

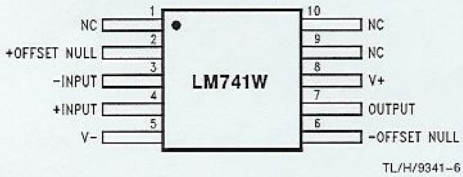
Dual-In-Line or S.O. Package



TL/H/9341–3

Order Number LM741J, LM741J/883, LM741CM, LM741CN or LM741EN
See NS Package Number J08A, M08A or N08E

Ceramic Flatpak



TL/H/9341–6

Order Number LM741W/883
See NS Package Number W10A

*LM741H is available per JM38510/10101

APPENDIX H.2: Data Sheet Of Op-Amp 741

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

(Note 5)

	LM741A	LM741E	LM741	LM741C
Supply Voltage	±22V	±22V	±22V	±18V
Power Dissipation (Note 1)	500 mW	500 mW	500 mW	500 mW
Differential Input Voltage	±30V	±30V	±30V	±30V
Input Voltage (Note 2)	±15V	±15V	±15V	±15V
Output Short Circuit Duration	Continuous	Continuous	Continuous	Continuous
Operating Temperature Range	−55°C to +125°C	0°C to +70°C	−55°C to +125°C	0°C to +70°C
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C	−65°C to +150°C	−65°C to +150°C
Junction Temperature	150°C	100°C	150°C	100°C
Soldering Information				
N-Package (10 seconds)	260°C	260°C	260°C	260°C
J- or H-Package (10 seconds)	300°C	300°C	300°C	300°C
M-Package				
Vapor Phase (60 seconds)	215°C	215°C	215°C	215°C
Infrared (15 seconds)	215°C	215°C	215°C	215°C
See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.				
ESD Tolerance (Note 6)	400V	400V	400V	400V

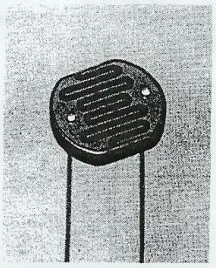
Electrical Characteristics (Note 3)

Parameter	Conditions	LM741A/LM741E			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	$T_A = 25°C$ $R_S \le 10\ k\Omega$ $R_S \le 50\Omega$		 0.8	 3.0		1.0	5.0		2.0	6.0	mV mV
	$T_{AMIN} \le T_A \le T_{AMAX}$ $R_S \le 50\Omega$ $R_S \le 10\ k\Omega$			4.0			 6.0			 7.5	mV mV
Average Input Offset Voltage Drift				15							µV/°C
Input Offset Voltage Adjustment Range	$T_A = 25°C, V_S = \pm 20V$	±10				±15			±15		mV
Input Offset Current	$T_A = 25°C$		3.0	30		20	200		20	200	nA
	$T_{AMIN} \le T_A \le T_{AMAX}$			70		85	500			300	nA
Average Input Offset Current Drift				0.5							nA/°C
Input Bias Current	$T_A = 25°C$		30	80		80	500		80	500	nA
	$T_{AMIN} \le T_A \le T_{AMAX}$			0.210			1.5			0.8	µA
Input Resistance	$T_A = 25°C, V_S = \pm 20V$	1.0	6.0		0.3	2.0		0.3	2.0		MΩ
	$T_{AMIN} \le T_A \le T_{AMAX}$, $V_S = \pm 20V$	0.5									MΩ
Input Voltage Range	$T_A = 25°C$							±12	±13		V
	$T_{AMIN} \le T_A \le T_{AMAX}$				±12	±13					V
Large Signal Voltage Gain	$T_A = 25°C, R_L \ge 2\ k\Omega$ $V_S = \pm 20V, V_O = \pm 15V$ $V_S = \pm 15V, V_O = \pm 10V$	 50			 50	 200		 20	 200		 V/mV V/mV
	$T_{AMIN} \le T_A \le T_{AMAX}$, $R_L \ge 2\ k\Omega$, $V_S = \pm 20V, V_O = \pm 15V$ $V_S = \pm 15V, V_O = \pm 10V$ $V_S = \pm 5V, V_O = \pm 2V$	 32 10			 25			 15			 V/mV V/mV V/mV

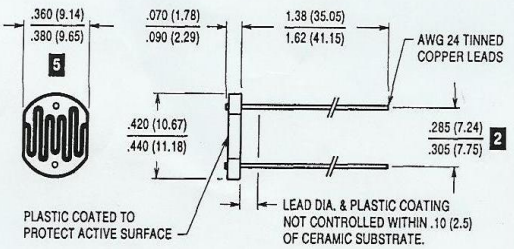
#

APPENDIX I: Data Sheet Of LDR Used in Light Sensor

PHOTOCONDUCTIVE CELL — VT400 Series



PACKAGE DIMENSIONS inch (mm)



ABSOLUTE MAXIMUM RATINGS

PARAMETER	SYMBOL	RATING	UNITS
CONTINUOUS POWER DISSIPATION	P_D	400	mW
DEMAND (20 MINUTES)		600	mW
DERATE ABOVE 25° C	$\Delta P_D / \Delta T$	8.0	mW/°C
TEMPERATURE RANGE			
OPERATING AND STORAGE	T_A	– 40 to +75	°C

ELECTRO-OPTICAL CHARACTERISTICS @ 25° C (16 HRS. LIGHT ADAPT, MIN.) 4

Part Number *	Resistance (Ohms) 3 6						Material Type	Sensitivity (γ, typ.) LOG (R10/R100) / LOG (100/10)	Maximum Voltage (V, pk)	Response Time @ 1 fc (ms, typ.)	
	1 fc 6500 K			2 fc 2850 K	DARK						
	Min.	Typ.	Max.	Typ.	Min.	sec.				Rise (1-1/e)	Fall (1/e)
VT43N1	4.0 k	8.0 k	12 k	—	300 k	30	3	0.90	250	90	18
VT43N2	8.0 k	16 k	24 k	—	300 k	30	3	0.90	250	90	18
VT43N3	16 k	32 k	48 k	—	500 k	30	3	0.90	400	90	18
VT43N4	33 k	66 k	100 k	—	500 k	30	3	0.90	400	90	18

EG&G VACTEC 10900 PAGE AVENUE ST. LOUIS, MO 63132 USA TEL: 314-423-4900 FAX: 314-423-3956 WEB: www.egginc.com/optogrp

Un

APPENDIX J: Data Sheet Of Microphone Used in Sound Sensor

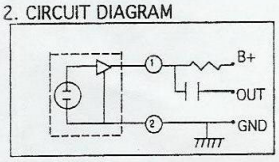
Rapid Electronics Ltd
Heckworth Close
Severalls Industrial Estate
Colchester Essex CO4 4TB
Telephone 01206 751166
Fax 01206 751188

DATA SHEET **Order code** 35-0190

1. SENSITIVITY
(ODB=1V/μ bar, 1KHZ,
VCC=4.5V, RL=1K Ω)

-60+/-3DB

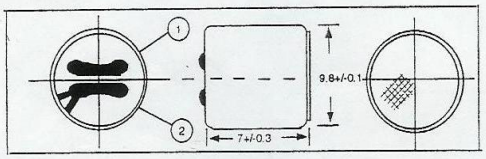
2. CIRCUIT DIAGRAM



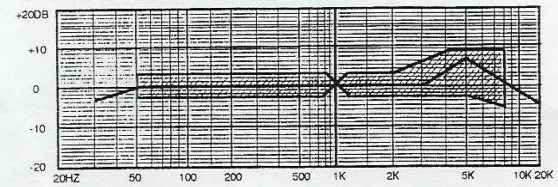
3. SPECIFICATIONS

1. IMPEDANCE: LOW
2. STANDARD VOLTAGE: 4.5 V
3. RANGE OF OPERATING VOLTAGE: 1.5 V TO 10 V
4. CURRENT DRAIN: 0.5mA Max
5. S/N RATIO: 40 db or more
6. MAXIMUM INPUT SOUND PRESSURE: 120 db SPL

4. DIMENSION



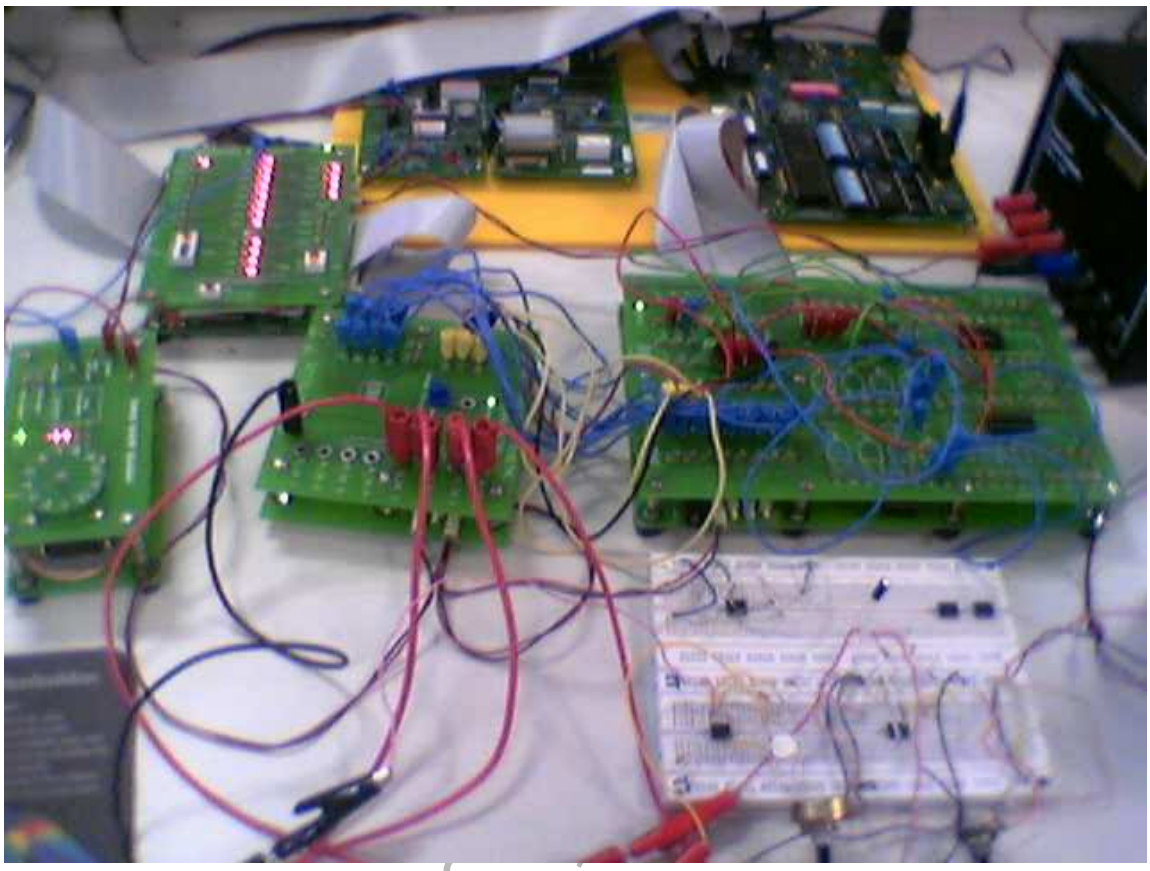
5. TYPICAL FREQUENCY RESPONSE CURVE



MAIN USAGE OF ECM UNITS: MICROPHONE, CASSETTE TAPE RECORDER, SONIC CONTROLLED TOY, INTERCOM SYSTEM, SONIC CONTROLLED SWITCH, TELEPHONE SET, DISCO LIGHT & ANSWERING MACHINE, ETC.

Un

APPENDIX K Practical laboratory set up of the project



University of

University of Hertfordshire